



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11)

**EP 1 050 821 B1**

(12)

**EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention  
of the grant of the patent:  
**25.10.2006 Bulletin 2006/43**

(51) Int Cl.:  
**G06F 12/14** <sup>(2006.01)</sup> **G11C 16/22** <sup>(2006.01)</sup>  
**G11C 7/16** <sup>(2006.01)</sup>

(21) Application number: **00302955.0**

(22) Date of filing: **07.04.2000**

(54) **Memory units, data processing units, and methods therefor**

Speichereinheiten, Datenverarbeitungseinheiten und zugehörige Verfahren

Unités de mémoire, unités de traitement de données et procédés correspondants

(84) Designated Contracting States:  
**DE FR GB**

(30) Priority: **07.04.1999 JP 9994999**  
**24.06.1999 JP 17818899**

(43) Date of publication of application:  
**08.11.2000 Bulletin 2000/45**

(73) Proprietor: **SONY CORPORATION**  
**Tokyo 141 (JP)**

(72) Inventors:  
• **Okaue, Takumi,**  
**Intellectual Property Department**  
**Tokyo 141 (JP)**  
• **Kihara, Nobuyuki,**  
**Intellectual Property Department**  
**Tokyo 141 (JP)**

• **Yokota, Teppei,**  
**Intellectual Property Department**  
**Tokyo 141 (JP)**

(74) Representative: **Pilch, Adam John Michael et al**  
**D Young & Co**  
**120 Holborn**  
**London EC1N 2DY (GB)**

(56) References cited:  
**US-A- 5 491 774** **US-A- 5 749 088**

• **TING T-K J ET AL: "A 50-NS CMOS 256K**  
**EEPROM" IEEE JOURNAL OF SOLID-STATE**  
**CIRCUITS, IEEE INC. NEW YORK, US, vol. 23, no.**  
**5, 1 October 1988 (1988-10-01), pages 1164-1170,**  
**XP000037032 ISSN: 0018-9200**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

## Description

[0001] The present invention relates generally to data security, and more particularly to memory units, data processing units, and methods therefor, such as in the case of a memory card, which is removably attachable to a data processing unit and which includes a data security means.

[0002] In conventional non-volatile memory such as EEPROM (Electrically Erasable Programmable ROM), two transistors are employed to store one bit of information. As a result, the memory area per bit is large, which limits the ability to raise the integration of the memory. On the other hand, this problem has been eliminated in a recently-developed flash memory in which one bit is stored using a single transistor according to the "all-bit-simultaneous-erase" method. In the not so distant future, it is expected that flash memories will replace conventional record mediums such as magnetic and optical discs in many applications.

[0003] Flash memory-based memory cards or "memory sticks<sup>TM</sup>" that are removably attachable to a card reading/recording unit are also known. With the advent of this type of memory card, digital audio recording/reproducing units have been developed which use the memory card instead of a conventional disc shaped medium such as a CD (Compact Disc) or a mini-disc.

[0004] An audio recorder that uses a memory card as a record medium typically employs a data compressing method which allows data to be restored in a relatively high quality for recording/reproducing. Encryption techniques can be implemented to protect the copyright of music titles recorded and reproduced with this audio recorder. As an example, the audio recorder can be designed to determine, via an encryption technique, whether a memory card is invalid and thus prohibited from being used with the recorder. In other words, a valid recorder and a valid memory card in combination allow encrypted data to be decrypted. In addition to the copyright protection, encryption technologies may be used to protect the security of other information stored in the memory card.

[0005] Conventional memory cards do not have an encrypting function. Thus, when secret data is recorded to a memory card, the data is encrypted on the "set" side, i.e., in the device ("set") that the card is inserted into and which sets up the data for recording. The encrypted data is then transferred to the memory card for storage. If a decryption key is also stored in the memory card, the data security of the card is compromised. On the other hand, when a decryption key is stored in a particular set, data originally encrypted by that set and recorded on a memory card cannot be decrypted by sets other than that particular set. Thus, the compatibility of memory cards cannot be maintained. To solve this problem, a system has been proposed in which a set and a memory card each have an encrypting function, thus enabling the set and memory card to be mutually authenticated. The memory card in this case can be considered a "smart card" having processing circuitry to carry out the data encryption. With this approach, both the security and compatibility of cards can be maintained.

[0006] A security unit having the above authenticating and encrypting functions may encrypt according to the Data Encryption Standard (DES). The DES is a block encrypting system in which text is block-segmented and each block segment is encrypted. With DES, input data of 64 bits is encrypted with a key of 64 bits (in reality, a key of 56 bits and a parity of 8 bits) and encrypted data of 64 bits is output. The DES has four use modes, one of which is a Cipher Block Chaining (CBC) mode. The CBC mode is a feedback type mode in which text of 64 bits and the preceding encrypted data (of 64 bits) are XORed and the result is input to the DES unit. In the initial state, since there is no encrypted data, an initialization vector is used. In addition, as data is being exchanged between the set and the memory card, random numbers may be generated and added to the data.

[0007] There are many applications in which non-copyrighted data is recorded to a memory card and reproduced therefrom. Examples include the recording of conversational speech (which is typically compressed with a high compression ratio prior to storing the same), image data from an electronic still camera or a video camera, and so forth. In these cases it is unnecessary to provide a means for protecting the copyright of the data. Generally, a security type memory card having an encrypting function is more costly than a non-security type memory card (namely, a conventional memory card). Thus, security type memory cards (and associated sets) are used in applications that require it, while non-security type cards and sets may be used for other applications to reduce cost. Prior art security type sets are usable only with security type memory cards, whereas non-security type sets are usable only with non-security type memory cards.

[0008] TING T-K ET AL: "A 50NS CMOS 256K EEPROM" IEEE JOURNAL OF SOLID-STATE CIRCUITS, IEEE INC. NEW YORK, US, vol. 23, no. 5, 1 October 1988 (1988-10-01), pages 1164-1170, XP000037032 ISSN: 0019-9200 discloses an EEPROM that includes a memory array, programming control logic, and a data protection fuse. During a write operation, 8 data bits and 4 parity bits are loaded into column latches and then written in the memory array. The programming control logic generates control signals to latch the address and data. Page-mode programming is employed. The page-mode cycle includes a loading period and a programming period. A data protection mode can be set by issuing a three code instruction during the loading period of the page-mode cycle. Once the data protection mode is set, no further programming can be accomplished unless the same three-code instruction is issued at the beginning of each loading period. A six-code instruction is used to reset the data-protection mode, which is controlled by an EEPROM

fuse. The circuit is designed to prevent repetitive fuse programming and thus maintain maximum endurance of the fuse.  
**[0009]** United States Patent No US-A-5 491 774 discloses a memory unit according to the pre-characterising part of claim 1 hereof.

**[0010]** The present invention provides a memory unit as set forth in claim 1 hereof, a data processing system as set forth in claim 9 hereof, and a data processing method as set forth in claim 10 hereof.

**[0011]** Thus, the memory unit is capable of being used with either a security-type or non-security type data processing unit ("set"). A non-security type set only transmits first control data, which is used to store and/or retrieve data (which is non-encrypted) to/from the memory unit. The security type set protects the security of data stored within the non-volatile memory by transmitting both first and second control data to the memory unit. Accordingly, the non-security type data processing unit can operate with both security-type and non-security type memory cards. Consequently, the compatibility of a security type memory unit can be improved.

**[0012]** The present invention recognises that, from a view point of compatibility, it would be desirable in some applications for a security type memory card to be usable with a non-security type set, e.g., a conventional set. In these applications, such as recording image data from a portable video recorder, the encrypting function of the memory card would not be used. To this end, the present invention provides a security-type memory unit that can be used with both security-type and non-security type data processing units (sets).

**[0013]** The invention will now be described by way of example with reference to the accompanying drawings, throughout which like parts are referred to by like references, and in which:

FIG. 1 depicts the overall structure of a recorder/player and a memory card in accordance with an embodiment of the present invention;

FIG. 2 depicts the internal structure of a security type memory card in accordance with an embodiment of the present invention;

FIG. 3 depicts the internal structure of a non-security type memory card;

FIG. 4 depicts the structure of a file system processing hierarchy of a flash memory according to an embodiment of the present invention;

FIG. 5 illustrates a format of a physical data structure of a flash memory;

FIG. 6 depicts the structure of a boot block of a flash memory;

FIG. 7 depicts the structure of boot and attribute information of a boot block of a flash memory;

FIGS. 8A and 8B illustrate the relation between contents and a key;

FIG. 9 is a diagram to which reference will be made in explaining an encrypting process in a record operation;

FIG. 10 is a diagram to which reference will be made in explaining an authenticating process;

FIG. 11 is a diagram to which reference will be made in explaining an encrypting process in a record operation;

FIG. 12 is a diagram to which reference will be made in explaining an encrypting process in a reproducing operation;

FIG. 13 is a diagram to which reference will be made in explaining an encrypting process in a reproducing operation;

FIG. 14 is a diagram to which reference will be made in explaining an operation of an interface disposed between the recorder and the memory card;

FIG. 15 is a diagram to which reference will be made in explaining an operation of an interface disposed between the recorder and the memory card;

FIG. 16 is a table depicting examples of protocol commands that may be used in embodiments of the invention;

FIGS. 17-18 are tables illustrating commands that may be used in embodiments of the invention; and

FIG. 19 is a schematic block diagram of a memory unit in accordance with the invention.

**[0014]** FIG. 1 is a block diagram showing the structure of a digital audio recorder/player 1 according to a preferred embodiment of the present invention. Digital audio recorder/player 1 records and reproduces a digital audio signal using a detachable memory card (or a Memory Stick™) 40. Recorder/player 1 may be a part of an audio system along with an amplifying unit (not shown), speakers (not shown), a CD player (not shown), an MD recorder (not shown), a tuner (not shown), and so forth. However, it should be noted that the present invention may be applied to other audio sets. For instance, recorder/player 1 may be a portable device. The present invention may also be applied to a set top box that records digital audio data that is circulated via satellite data communication, digital broadcast, or the Internet, etc. Moreover, the present invention may be applied to a system that records/reproduces moving picture data and still picture data rather than audio data. A system according to an embodiment of the present invention may also record and reproduce additional information, such as picture and text, other than a digital audio signal.

**[0015]** Recorder/player 1 (which can also be considered a "data processing unit") has a Central Processing Unit (CPU) 2, a security block 3, an operation button 4, and a display device 5. Security block 3 includes a Data Encryption Standard ("DES") encrypting circuit. Data such as a record command, a reproduction command, or the like corresponding to a user's operation of operation button 4 is supplied to CPU 2 through bus 16. Various information, the operation state of recorder/

player 1, and so forth are displayed on display device 5. An audio interface 6 is disposed between an external input/output, which will be described in further detail below, and an internal audio encoder/decoder7.

**[0016]** As will be described later, memory card 40 is an IC chip having a flash memory (non-volatile memory) 42, a control block 41, a security block 52 (security block 52 may include a DES encrypting circuit), a communication interface, a register, and so forth. Memory card 40 is attachable to recorder/player 1 and detachable therefrom. The recorder/player 1 is also compatible with a memory card that does not have an encrypting function (namely, security block 52).

**[0017]** Audio encoder/decoder7 encodes digital audio data in accordance with a highly efficient encoding method to be written to memory card 40. In addition, encoder/decoder 7 decodes encoded data read from memory card 40. The highly efficient ATRAC3 format encoding method, which is a modification of the Adaptive Transform Acoustic Coding ("ATRAC") format used for MDs, may be used.

**[0018]** In the ATRAC3 format, audio data sampled at 44.1 kHz and quantized with 16 bits is encoded with high efficiency. The minimum data unit of audio data for processing is a sound unit ("SU"). 1 SU contains data of 1024 samples, thus comprising (1024 x 16 bits x 2 channels) bits, that is compressed to data of several hundred bytes. The duration of 1 SU is approximately 23 msec. Under this highly efficient encoding method, the size of compressed data is approximately 10 times smaller than that of the original data. As compared to the ATRAC1 format used in MDs, an audio signal compressed and decompressed according to the ATRAC3 format is less deteriorated in audio quality.

**[0019]** Illustratively, an analog input 8 supplies a reproduction output signal of an MD, a tuner, or a tape to an Analog-to-Digital("A/D") converter 9. A/D converter 9 converts the signal from analog input 8 to a digital audio signal (sampling frequency = 44.1 kHz; the number of quantizing bits = 16) and supplies the converted digital audio signal to audio interface 6. A digital input 10 supplies a digital output signal of an MD, a CD, a digital broadcast signal, or network circulated audio data to audio interface 6. The digital input signal is transmitted through, for example, an optical cable. Audio interface 6 selects an input digital audio signal from A/D converter 9 and digital input 10 and supplies the selected input digital audio signal to audio encoder/decoder7.

**[0020]** Audio encoder/decoder7 encodes the input digital audio signal and supplies the encoded data to security block 3. Security block 3 encrypts the encoded data received from audio encoder/decoder 7 so as to protect copyrights on the contents of said data (in this example, a digital audio signal). Security block 3 of recorder/player 1 may have a plurality of master keys and a unit unique storage key. In addition, security block 3 may have a random number generating circuit (not shown). When memory card 40 having security block 52 is attached to recorder/player 1, security block 3 of recorder/player 1 determines whether or not memory card 40 is valid (namely, authenticates memory card 40). After security block 3 of recorder/player 1 has properly authenticated memory card 40, security block 3 of recorder/player 1 and security block 52 of memory card 40 share a session key.

**[0021]** The encrypted audio data that is output from security block 3 is supplied to CPU 2. CPU 2 communicates with memory card 40 through a bidirectional serial interface 11. In an embodiment, memory card 40 is attached to an attaching/detaching mechanism (not shown) of recorder/player 1. CPU 2 writes the encrypted data to flash memory 42 of memory card 40. The encrypted data is serially transmitted between CPU 2 and memory card 40.

**[0022]** CPU 2 reads encrypted audio data from memory card 40 through memory interface 11 and supplies such data to security block 3. Security block 3 decrypts the encrypted audio data. The decrypted audio data is supplied to audio encoder/decoder 7 which decodes the decrypted audio data. An output signal of audio encoder/decoder 7 is supplied to a D/A converter 12 through audio interface 6. D/A converter 12 converts the digital audio data into an analog audio signal and transmits the same through output 13. Audio data received from audio encoder/decoder 7 and decrypted data received from security block 3 may also be outputted as digital output signals through outputs 14 and 15, respectively, through interface 6.

**[0023]** FIG. 2 is a block diagram showing the internal structure of memory card 40. Memory card 40 is a one chip integrated circuit ("IC") comprising control block 41, security block 52, and flash memory 42. As shown in FIG. 2, bidirectional serial interface 11 disposed between CPU 2 of recorder/player 1 and memory card 40 is composed of 10 lines, which include a clock line SCK for transmitting the clock signal that is transmitted along with data, a status line SBS for transmitting a status signal, a data line DIO for transmitting data, an interrupt line INT, two GND lines, two VCC lines, and two reserved lines.

**[0024]** Four major lines of the 10 lines are clock line SCK, status line SBS, data line DIO, and interrupt line INT. Clock line SCK is used to send a clock signal to synchronize data transfer. Status line SBS is used to send a status signal that represents the status of memory card 40. Data line DIO is used to input and output a command and encrypted audio data. Interrupt line INT is used to send an interrupt request signal from memory card 40 issues to CPU 2 of recorder/player 1. When memory card 40 is attached to recorder/player 1, an interrupt signal is generated. In another embodiment, the interrupt signal is sent through data line DIO in which case interrupt line INT is grounded and not used.

**[0025]** A serial/parallel and parallel/serial interface block ("S/P and P/S IF block") 43 is an interface of control block 41 coupled to interface 11. S/P and P/S IF block 43 converts serial data received from recorder/player 1 into parallel data. It also converts parallel data of control block 41 into serial data, and supplies the serial data to recorder/player 1. In addition, S/P and P/S IF block 43 separates a command and data received through data line DIO into those for

accessing flash memory 42 and those for performing an encrypting process.

**[0026]** In other words, with the data line DIO, after a command is sent, data is sent. S/P and P/S IF block 43 determines whether the received command and data are for accessing flash memory 42 or for performing the encrypting process by the code of the received command. Corresponding to the determined result, a command for accessing flash memory 42 is stored to a command register 44 and data is stored to a page buffer 45 and a write register 46. In association with write register 46, an error correction code encoding circuit 47 is disposed. Error correction code encoding circuit 47 generates a redundant code of an error correction code for data temporarily stored in page buffer 45.

**[0027]** Output data of command register 44, page buffer 45, write register 46, and error correction code encoding circuit 47 is supplied to a flash memory interface and sequencer ("memory IF and sequencer") 51. Memory IF and sequencer 51 is an interface coupled to flash memory 42 and controls data exchanged between flash memory 42 and control block 41, for example, data is written to flash memory 42 through memory IF and sequencer 51.

**[0028]** Data read from flash memory 42 is supplied to page buffer 45, a read register 48, and an error correcting circuit 49 through memory IF and sequencer 51. Error correcting circuit 49 corrects an error(s) of data stored in page buffer 45. Error corrected data output from page buffer 45 and data output from read register 48 are supplied to S/P and P/S IF block 43 and then supplied to CPU 2 of recorder/player 1 through serial interface 11.

**[0029]** To protect copyrights on the contents (audio data compressed in the ATRAC3 format ("ATRAC3 data")) written to flash memory 42, security block 3 of recorder/player 1 and security block 52 of memory card 40 cooperate to encrypt the contents. Security block 52 has a buffer memory 53, a DES encrypting circuit 54, a non-volatile memory 55, and so forth.

**[0030]** As shown in FIG. 2, a configuration ROM 50 is disposed in control block 41. Configuration ROM 50 stores version information and various kinds of attribute information of memory card 40. Memory card 40 has a write protection switch 60 operable by a user. When switch 60 is placed in a write protection position, even if recorder/player 1 sends an erase command to flash memory 42, data stored in flash memory 42 is prohibited from being erased. When switch 60 is placed in a non-write protection position, data stored in flash memory 42 is erasable. An oscillator 61 generates a clock signal used as a timing reference for processes performed in memory card 40.

**[0031]** Security block 52 of memory card 40 has a plurality of authentication keys and a memory card unique storage key. Non-volatile memory 55 stores a decryption or storage key that cannot be accessed from outside of security block 52. Security block 52 has a random number generating circuit. Security block 52 can authenticate recorder/player 1 (which may form a dedicated system that uses a predetermined data format) and share a session key therewith. A contents key for encrypting ATRAC3 data is encrypted with the session key and sent between recorder/player 1 and memory card 40. As with security block 52 of memory card 40, security block 3 of recorder/player 1 has a set unique storage key. When contents have been encrypted and are to be stored to flash memory 42, a corresponding contents key is encrypted using the storage key and stored with the encrypted contents.

**[0032]** FIG. 3 shows a memory card 40' that does not have an encrypting function. In other words, memory card 40' is a non-security type memory card. Unlike memory card 40 shown in FIG. 2, memory card 40' does not include security block 52. The remaining structure of memory card 40' is substantially the same as that of memory card 40. In addition, the size and shape of memory card 40' may be the same as that of memory card 40. Since recorder/player 1 shown in FIG. 1 is a security type recorder, recorder/player 1 and the memory card 40 are mutually authenticated and a key is communicated therebetween. When memory card 40', shown in Fig. 3, is attached to recorder/player 1, recorder/player 1 determines that memory card 40' is a non-security type memory card and that it cannot be used with recorder/player 1.

**[0033]** There are several methods by which recorder/player 1 may determine the type of memory card attached thereto. As one example, when memory card 40' is attached to recorder/player 1, a key is sent from recorder/player 1 to memory card 40' so as to authenticate it. Since memory card 40' does not send a correct response to recorder/player 1, recorder/player 1 determines that memory card 40' is not of the security type after a time-out period. As another example, when memory card 40 or 40' is attached to recorder/player 1, identification information that represents whether or not the memory card is of the security type may be recorded in a predetermined area (boot area) of the memory card. Upon reading such identification information, recorder/player 1 can determine the type of memory card attached thereto.

**[0034]** In addition to recorder/player 1 shown in FIG. 1, a unit that can use non-security type memory card 40' is presented. One example is a digital "palm-corder" that records a picture photographed with a Charge Coupled Device ("CCD") camera to memory card 40' and reproduces the photographed picture therefrom. As will be described later, according to an embodiment of the present invention, to enhance the compatibility of memory card 40, it is structured so that a non-security device such as a digital palm-corder can record and reproduce data using memory card 40. In other words, as described above, S/P and P/S IF block 43 has a function for separating command and data for flash memory 42 and those for security block 52.

**[0035]** In accordance with an embodiment, memory cards 40 and 40' store data using the File Allocation Table ("FAT") file system of a personal computer as with a disc shaped recording medium. Flash memory 42 comprises an Initial Program Load ("IPL") area, a FAT area, and a route directory. The IPL area stores the address of a program that is initially loaded to a memory of recorder/player 1. In addition, the IPL area stores various kinds of information of flash

memory 42. The FAT area stores data with respect to memory blocks in flash memory 42. In other words, the FAT area stores values that represent non-used blocks, the next block number, bad blocks, and the last block. The route directory area stores a directory entry (file attribute, updated date (year, month, and day), start cluster, file size, and so forth).

**[0036]** In addition to the file management system defined in the format of memory cards 40 and 40', file management information (a track information management file) for a music file may be defined. The track information management file is stored in flash memory 42 using a user block of memory cards 40 and 40'. Thus, even if the FAT of memory card 40 or 40' is broken, the file can be restored.

**[0037]** The track information management file is created by CPU 2. When the power of recorder/player 1 is turned on, CPU 2 determines whether or not memory card 40 or 40' has been attached to recorder/player 1. When memory card 40 or 40' has been attached to recorder/player 1, CPU 2 reads a boot block of flash memory 42. In accordance with the identification information of the boot block, CPU 2 determines whether or not the attached memory card is a security type memory card.

**[0038]** If memory card 40 is attached (i.e., security type), CPU 2 performs an authenticating process. Other data read from memory card 40 is stored in a memory (not shown) managed by CPU 2. In flash memory 42 of memory card 40 or 40' that has not been used, before it is shipped, a FAT and a route direction are written. When data is recorded, the track information management file is created. After CPU 2 has authenticated memory card 40, recorder/player 1 records or reproduces an encrypted ATRAC3 data file.

**[0039]** When data is recorded, a record command that is issued corresponding to the operation of operation button 4 is sent to CPU 2. The input audio data is compressed by encoder/decoder 7. The ATRAC3 data received from encoder/decoder 7 is encrypted by security block 3. CPU 2 stores the encrypted ATRAC3 data to flash memory 42 of memory card 40. Thereafter, the FAT and the track information management file are updated. Whenever the file is updated (namely, after audio data is recorded), the FAT and the track information management file are rewritten to a memory controlled by CPU 2. When memory card 40 is detached from recorder/player 1 or the power of recorder/player 1 is turned off, the final FAT and the track information management file are supplied from the memory to flash memory 42 of memory card 40. In this case, whenever audio data has been recorded, the FAT and the track information management file stored in flash memory 42 may be rewritten. When data is edited, the contents of the track information management file are updated.

**[0040]** FIG. 4 is a schematic diagram showing the hierarchy of the file system processes of a computer system that uses memory card 40 or 40' as a storage medium. As shown therein, the top hierarchical level is an application process layer. The application process layer is followed by a file management process layer, a logical address management layer, a physical address management layer, and a flash memory access layer. The file management process layer is the FAT file system. Physical addresses are assigned to individual blocks of flash memory 42 in memory card 40 or 40'. The relationship between the blocks of flash memory 42 and the physical addresses thereof does not vary. Logical addresses are addresses that are logically handled on the file management process layer.

**[0041]** FIG. 5 is a schematic diagram showing the physical structure of data handled in flash memory 42 of memory card 40 or 40'. In flash memory 42, a data unit (referred to as a segment) is divided into a predetermined number of blocks (fixed length). One block is divided into a predetermined number of pages (fixed length). In flash memory 42, data is erased one block at a time. Data is written to flash memory 42 or read therefrom one page at a time. The size of each block is the same. Likewise, the size of each page is the same. One block is composed of page 0 to page m. One block may have a storage capacity of 8 KB (kilobytes) or 16 KB and one page may have a storage capacity of 512 B (bytes). When one block has a storage capacity of 8 KB, the total storage capacity of flash memory 42 is 4 MB (512 blocks) or 8 MB (1024 blocks). When one block has a storage capacity of 16 KB, the total storage capacity of flash memory 42 is 16 MB (1024 blocks), 32 MB (2048 blocks), or 64 MB (4096 blocks).

**[0042]** One page is composed of a data portion of 512 bytes and a redundant portion of 16 bytes. The first three bytes of the redundant portion is an overwrite portion that is rewritten whenever data is updated. The first three bytes successively contain a block status area, a page status area, and an update status area. The remaining 13 bytes of the redundant portion are fixed data that depends on the contents of the data portion. The 13 bytes contain a management flag area (1 byte), a logical address area (2 bytes), a format reserve area (5 bytes), a dispersion information Error-Correcting Code ("ECC") area (2 bytes), and a data ECC area (3 bytes). The dispersion information ECC area contains redundant data for an error correction process for the management flag area, the logical address area, and the format reserve area. The data ECC area contains redundant data for an error correction process for the data in the 512-byte data portion.

**[0043]** The management flag area contains a system flag (1: user block, 0: boot block), a conversion table flag (1: invalid, 0: table block), a copy prohibition flag (1: copy allowed, 0: copy not allowed), and an access permission flag (1: free, 0: read protect).

**[0044]** The first two blocks - blocks 0 and 1 are boot blocks. Block 1 is a backup of block 0. The boot blocks are top blocks that are valid in memory card 40 or 40'. When memory card 40 or 40' is attached to recorder/player 1, the boot blocks are accessed first. The remaining blocks are user blocks. Page 0 of a boot block contains a header area, a system entry area, and a boot and attribute information area. Page 1 of a boot block contains a prohibited block data area. Page

2 of a boot block contains a CIS (Card Information Structure)/IDI (Identify Drive Information) area.

**[0045]** FIG. 6 shows the format of pages 0, 1, and 2 of a boot block. A header (368 bytes) of a boot block stores a boot block ID, a format version, and the number of valid entries of the boot block. A system entry (48 bytes) stores the start position of the prohibited block data, the data size thereof, the data type thereof, the data start position of CIS/IDI, the data size thereof, and the data type thereof. The boot and attribute information contains memory card type (read only type, rewritable type, or hybrid type), the block size, the number of blocks, the number of total blocks, the security/non-security type, the card fabrication data (date of fabrication), and so forth.

**[0046]** FIG. 7 shows the structure of the boot & attribute information (96 bytes) shown in FIG. 6. The boot & attribute information may include the class of the memory card, the type (read only, read write enable, hybrid of both types, etc.), the block size, the number of blocks, the total number of blocks, the security type/non-security type, the production data (the date of production: year, month, day), and so forth. Recorder/player 1 determines whether or not a memory card is of the security type using the security type information (one byte). In FIG. 7, (\*1) represents a data item that recorder/player 1 reads and checks when a memory card is attached thereto; and (\*2) represents production/quality management data item.

**[0047]** It is appreciated that the insulation film of flash memory 42 deteriorates whenever data stored therein is rewritten. Thus, the service life of memory card 40 or 40' is limited by the number of times flash memory 42 is rewritten. Accordingly, it is preferable to prevent a particular storage area (block) of flash memory 42 from being repeatedly accessed. Consequently, when data stored at a particular physical address is to be rewritten, updated data is not written back to the same block. Instead, the updated data is written to a block that has not been used. Thus, after data is updated, the relationship between physical addresses and logical addresses varies. When such a process (referred to as a swapping process) is performed, the same block is prevented from being repeatedly accessed. Thus, the service life of flash memory 42 can be prolonged.

**[0048]** Since a logical address corresponds to data written to a block, even if updated data is physically moved to another block, the same logical address may be maintained in the FAT. The swapping process causes the relationship between logical addresses and physical addresses to vary. Thus, a conversion table that converts logical addresses into physical addresses is changed accordingly when such a swapping process is performed. By referencing the conversion table, a physical address corresponding to a logical address designated by the FAT is obtained. Thus, the updated data can be properly accessed using the same logical address.

**[0049]** The logical address - physical address conversion table is stored in a memory Random Access Memory ("RAM") by CPU 2. However, when the storage capacity of the RAM is small, the logical address - physical address conversion table can be stored in flash memory 42. This table basically correlates logical addresses (two bytes) arranged in ascending order with physical addresses (two bytes). Since the maximum storage capacity of flash memory 42 is 128 MB (8192 blocks), with two bytes, 8192 addresses can be represented. In addition, the logical address-physical address conversion table is managed segment by segment. The size of the logical address - physical address conversion table is proportional to the storage capacity of flash memory 42. If the storage capacity of flash memory 42 is 8 MB (two segments), two pages corresponding to the two segments are used for the logical address - physical address conversion table. If the logical address - physical address conversion table is stored in flash memory 42, one bit of the management flag of the redundant portion of each page represents whether or not a relevant block has been stored in the logical address - physical address conversion table.

**[0050]** Next, the security protecting function will be further described. First of all, with reference to FIGS. 8A and 8B, the relation between a key and contents will be described. Each tune (or song) stored in flash memory 42 may be referred to as a track. FIG. 8A illustrates one track stored in flash memory 42. As shown in FIG. 8A, each track includes a key area (header) 101. A contents key CK created for each track (title) of encrypted audio data is encrypted with a memory card unique storage key Kstm and the resultant data is stored to key area 101. DES is used for an encrypting process for the contents key CK and the storage key Kstm. DES (Kstm, CK) represents that the contents key CK is encrypted with the storage key Kstm. An encoded value preferably has 64 bits composed of 56 bits of data and 8 bits of an error detection by Cyclical Redundancy Checking ("CRC").

**[0051]** Each track is divided into parts 102. A parts key PK is recorded with each part. Illustratively, the track shown in FIG. 8A comprises only one part 102. Part 102 is a set of blocks 103 (16 KB each). Each block 103 stores a block seed BK\_SEED and an initial vector INV. The part key PK is paired with a contents key CK so as to create a block key BK for encrypting the contents. In other words,  $BK = DES(CK (+) PK, BK\_SEED)$  (56 bits + 8 bits) (where (+) represents an exclusive-OR). The initial vector INV is an initial value for an encrypting/decrypting process for a block.

**[0052]** FIG. 8B relates to contents data in recorder/player 1. A contents key CK for each track of contents is decrypted and the resultant data is re-encrypted with a recorder/player unique storage key Kstd. The re-encrypted data is stored in a key area 111. In other words, the decrypting process is denoted by  $IDES(Kstm, CK)$  (56 bits + 8 bits). The re-encrypting process is denoted by  $DES(Kstd, CK)$  (56 bits + 8 bits). A part key PK for creating a block key BK is recorded for each part 112 of the contents. Each block 113 of a part 112 may store a block seed BK-SEED and an initial vector INV. As with the memory card, the block key BK is represented as  $BK = DES(CK (+) PK, BK\_SEED)$  (56 bits + 8 bits).

Write Operation to Memory Card 40

**[0053]** An encrypting process which may be utilized in a recording (write) operation of recorder/player 1 will now be explained with reference to FIG. 9. For simplicity, in FIG. 9, similar portions to those in FIG. 1 are denoted by similar reference numerals and their description is omitted. In addition, interface 11, bus 16, and control block 41, through which data and commands are transferred between the components of recorder/player 1 and memory card 40, have been omitted from FIG. 9 and the following process explanation for simplicity. In FIG. 9, SeK is a session key shared between recorder/player 1 and memory card 40 after they have been mutually authenticated. In FIG. 9, reference numeral 10' is a CD and a source of a digital audio signal inputted at digital input 10.

**[0054]** When memory card 40 is attached to recorder/player 1, recorder/player 1 determines whether or not memory card 40 is a security type memory card by use of the identification information in the boot area thereof. Since memory card 40 is a security type memory card, recorder/player 1 and memory card 40 are mutually authenticated.

**[0055]** The process of mutual authentication between recorder/player 1 and memory card 40 will be hereinbelow described with reference to FIG. 10.

**[0056]** After a write request signal is sent from recorder/player 1 to memory card 40, recorder/player 1 and memory card 40 mutually authenticate again, as will be described in further detail with reference to FIG. 10. If recorder/player 1 and memory card 40 recognize each other as legitimate in accordance with the mutual identification process, a key writing process, as will be described in further detail with reference to FIG. 11, is performed. Otherwise, the write operation is terminated. After the key writing process is complete, audio data is encrypted and written to memory card 40 through interface 11 by CPU 2.

**[0057]** With reference to FIG. 9, recorder/player 1 generates a random number for each track of data (tune) to be written and creates a corresponding contents key CK according to each of the random numbers. Security block 3 of recorder/player 1 encrypts contents key CK using session key SeK. Recorder/player 1 outputs the encrypted contents key CK to memory card 40. DES encrypting/decrypting circuit 54 of security block 52 in memory card 40 decrypts the encrypted contents key CK, and re-encrypts the decrypted contents key CK using a storage key Kstm from memory 55. Memory card 40 outputs the re-encrypted CK to recorder/player 1 (CPU 2). Recorder/player 1 (CPU 2) sets the re-encrypted contents key CK in the key area 111 (as shown in FIG. 8B) of each track. Recorder/player 1 generates a random number for each part data area 112 (as shown in FIG. 8B) of each track, and creates a part key PK according to each random number. Each created part key PK is set in a corresponding part data area 112 by CPU 2.

**[0058]** A temporary key TMK may be generated by performing an XOR of part key PK and contents key CK by recorder/player 1 for each part data area 112 as shown below in equation (1). The creation of temporary key TMK is not limited to using an XOR function. It is possible to use other functional operators, such as a simple AND operator.

$$TMK = PK \text{ XOR } CK \quad (1)$$

**[0059]** Recorder/player 1 generates a random number for each block 113 of each part data area 112 and creates block seed BK\_SEED according to each random number. Further, recorder/player 1 (CPU 2) sets the created block seed BK\_SEED into its proper position in each corresponding block 113. Recorder/player 1 uses the temporary key TMK and the block seed BK\_SEED in equation (2) to perform a Message Authentication Code ("MAC") operation to create block key BK for each block 113.

$$BK = MAC(TMK, BK\_SEED) \quad (2)$$

**[0060]** It is possible to perform processing other than a MAC operation by using a secret key on the input of a SHA-1 (secure Hash algorithm), RIPEMD-160, or other one-way Hash functions to create block key BK. Here, the one-way function  $f$  defines a function from which it is easy to calculate  $y = f(x)$  from  $x$ , but conversely difficult to find  $x$  from  $y$ . A one-way Hash function is described in detail in the "Handbook of Applied Cryptography, CRC Press".

**[0061]** Audio encoder/decoder 7 compresses the digital audio signal inputted to digital input 10 from CD 10' or the digital signal from A/D converter 9, which converts an analog audio signal inputted to analog input 8 into a digital signal, in accordance with the ATRAC3 format. Then, security block 3 encrypts the compressed audio data in the Cipher Block Chaining ("CBC") mode by using the block key BK, the CBC mode being a data encryption mode prescribed in Federal Information Processing Standard ("FIPS") PUB 81 ("DES MODES OF OPERATION").

**[0062]** Recorder/player 1 adds headers to the encrypted audio data and outputs the results to memory card 40. Memory card 40 writes the encrypted audio data and headers into flash memory 42. At this point, writing of audio data from



recorder/player to memory card 40 is complete.

**[0063]** FIG. 10 shows an authenticating process performed between recorder/player 1 (SET) and memory card 40 (MEMORY CARD). At step S1, the random number generator of security block 52 in memory card 40 generates a random number  $R_m$  and sends the random number  $R_m$  and the serial number ID of memory card 40 to recorder/player 1.

**[0064]** At step S2, recorder/player 1 receives  $R_m$  and ID and generates an authentication key  $IK_j$  according to the relationship  $IK_j = \text{MAC}(MK_j, ID)$ , where  $MK_j$  is one of the master keys stored in security block 3. Recorder/player 1 generates a random number  $R_d$  and creates a message authenticator  $MAC_A$  (Message Authentication Code) with the authentication key, namely,  $MAC(IK_j, R_d // R_m // ID)$ . Thereafter, recorder/player 1 generates a random number  $S_d$  and sends  $R_d // S_d // MAC_A // j$  to memory card 40.

**[0065]** At step S3, memory card 40 receives the data  $R_d // S_d // MAC_A // j$ , finds an authentication key  $IK_j$  from security block 52 corresponding to  $j$ , and calculates a  $MAC_B$  with the authentication key  $IK_j$  using  $R_d$ ,  $R_m$ , and ID. When the calculated  $MAC_B$  is equal to the received  $MAC_A$ , memory card 40 determines that recorder/player 1 is valid (i.e., authorized). At step S4, memory card 40 creates  $MAC_C = \text{MAC}(IK_j, R_m // R_d)$  and generates a random number  $S_m$ . Thereafter, memory card 40 sends  $S_m // MAC_C$  to recorder/player 1.

**[0066]** At step S5, recorder/player 1 receives  $S_m // MAC_C$  from memory card 40. Recorder/player 1 calculates  $MAC_D$  using  $IK_j$ ,  $R_m$ , and  $R_d$ . When the calculated  $MAC_D$  is equal to the received  $MAC_C$ , recorder/player 1 determines that memory card 40 is valid (i.e., authorized). At step S6, recorder/player 1 designates  $MAC(IK_j, R_m // R_d)$  as the sessionkey  $SeK$ . At step S7, memory card 40 designates  $MAC(IK_j, R_m // R_d)$  as the session key  $SeK$ . When recorder/player 1 and memory card 40 are mutually authenticated, the session key  $SeK$  is shared between them. The session key  $SeK$  is created whenever authentication is successful.

**[0067]** FIG. 11 shows a key writing process in the case that recorder/player 1 (SET) records audio data to flash memory 42 of memory card 40 (MEMORY CARD). At step S11, recorder/player 1 generates a random number for each track of contents and creates a contents key  $CK$ . At step S12, recorder/player 1 encrypts the contents key  $CK$  with the session key  $SeK$  and sends encrypted DES ( $SeK, CK$ ) to memory card 40.

**[0068]** At step S13, memory card 40 receives the data DES ( $SeK, CK$ ) from recorder/player 1 and decrypts the contents key  $CK$  with the session key  $SeK$ . The decrypting process is denoted by IDES ( $SeK, \text{DES}(SeK, CK)$ ). At step S14, memory card 40 re-encrypts the decrypted contents key  $CK$  with the storage key  $K_{stm}$  from memory 55 and sends the re-encrypted contents key DES ( $K_{stm}, CK$ ) to recorder/player 1.

**[0069]** At step S15, recorder/player 1 places the re-encrypted contents key  $CK$  in the key area 111 for managing the corresponding part data area 112 and performs a formatting process so that the re-encrypted contents key  $CK$  and the contents are recorded to flash memory 42 of memory card 40. To encrypt the contents, the contents key  $CK$  and the part key  $PK$  are exclusive-Ored (XOR, or alternatively, AND), as illustrated in Fig. 9 and equation 11 above. The result of the XOR operation is the temporary key  $TMK$ . The temporary key  $TMK$  is stored only in security block 3. Thus, the temporary key  $TMK$  is not accessible from outside of security block 3. At the beginning of each block 113, a random number is generated as a block seed  $BK\_SEED$ . The random number is stored in each part data area 112. Recorder/player 1 encrypts the block seed  $BK\_SEED$  with the temporary key  $TMK$  to obtain a block key  $BK$ . In other words, the relation of  $BK = (CK (+) PK, BK\_SEED)$  is obtained. The block key  $BK$  is stored only in security block 3. Thus, the block key  $BK$  is not accessible from outside of security block 3.

**[0070]** At step S16, recorder/player 1 encrypts the data in each part data area 112 block by block with the block key  $BK$  and sends the encrypted data and the data in key area 111 to memory card 40. Memory card 40 records the encrypted data and the data in key area 111 (header data) received from recorder/player 1 to flash memory 42 at step S17.

#### Read Operation from Memory card 40

**[0071]** A decrypting process for use in a reproducing (read) operation of recorder/player 1 will now be explained with reference to FIG. 12. For simplicity, in Fig. 12, similar portions to those in FIG. 1 are denoted by similar reference numerals and their description is omitted. In addition, interface 11, bus 16, and control block 41, through which data and commands are transferred between the components of recorder/player 1 and memory card 40, have been omitted from FIG. 12 and the following process explanation for simplicity.

**[0072]** A read request signal specifying a desired track of data (tune) is sent from recorder/player 1 to memory card 40. Recorder/player 1 and memory card 40 perform a mutual authentication operation, as above described with reference to FIG. 10. If recorder/player 1 and memory card 40 recognize each other as legitimate in accordance with the mutual identification process, a key writing process, as above described with reference to Fig. 11, is performed. Otherwise, the read operation is terminated. After the key writing process is complete, encrypted audio data is read from memory card 40 to recorder/player 1 by CPU 2.

**[0073]** Since mutual identification is carried out between memory card 40 and recorder/player 1, the encrypted contents key  $CK$  can be decrypted using the proper session key  $ScK$  only when memory card 40 and recorder/player 1 identify each other as legitimate. Therefore, illicit utilization of the audio data is easily avoided. Data read during the read operation

had been written by the above-described write operation shown in FIG. 9. The setting of the contents key CK and the part key PK in each part data area 112, and the block seed BK\_SEED in each block 113 is used for writing data to, and thus reading data from, the corresponding part data area 102. After step S6 of Fig. 10 is completed, memory card 40 and recorder/player 1 share session key SeK. The reading of audio data from memory card 40 proceeds as follows.

**[0074]** Memory card 40 specifies the data in the part data area 102 (FIG. 8A) corresponding to the read request signal and outputs the audio data in sound units SUs from the blocks 103 (FIG. 8A) in the specified part data area 102. Memory card 40 also reads the corresponding key area 101 (FIG. 8A) of the audio data and outputs it to recorder/player 1.

**[0075]** Recorder/player 1 picks-up the encrypted contents key CK from the data in the key area 101 and outputs it to memory card 40. DES encrypting/decrypting circuit 54 of security block 52 in memory card 40 decrypts the encrypted contents key CK using storage key Kstm stored in memory 55, and re-encrypts the decrypted contents key CK using session key SeK.

**[0076]** Memory card 40 outputs the re-encrypted contents key CK to recorder/player 1. Recorder/player 1 decrypts the re-encrypted contents key CK from memory card 40 using session key SeK. Recorder/player 1 then obtains the XOR of the decrypted contents key CK and the part key PK from data in each part data area 102 so as to obtain the temporary key TMK in accordance with equation (3).

$$TMK = PK \text{ XOR } CK \quad (3)$$

**[0077]** Recorder/player 1 uses the temporary key TMK and the block seed BK\_SEED in each part data area 102 to perform the MAC operation shown in the following equation (4) so as to obtain the block key BK. The block key BK is found for every block 103 as follows.

$$BK = MAC (TMK, BK\_SEED) \quad (4)$$

**[0078]** Security block 3 of recorder/player 1 decrypts the audio data by using the block key BK. More specifically, the audio data is decrypted for every block 103 using the individually found block key BK. Further, decryption is carried out in the same 16KB blocks 103 as used for encryption. Audio encoder/decoder7 expands the decrypted audio data according to the ATRAC3 system and outputs the decoded signal through digital output 14 or D/A converter 12 converts the digital audio signal into an analog signal and outputs the result through analog output 13. Alternatively, the ATRAC3 audio data from security block 3 is outputted through output 15. Audio encoder/decoder 7 expands the audio data in sound units SUs.

**[0079]** FIG. 13 shows the decrypting process when recorder/player 1 reproduces an audio track stored in flash memory 42 of memory card 40. As with the write operation shown in FIGS. 9 to 11, the session key SeK is shared between recorder/player 1 and memory card 40 after they are mutually authenticated.

**[0080]** At step S21, recorder/player 1 (SET) reads data from memory card 40 (MEMORY CARD) and obtains the contents key CK encrypted with the storage key Kstm (namely, DES (Kstm, CK)) and encrypted contents (part data area (s) 102 of the desired track). Thereafter, recorder/player 1 sends the contents key CK encrypted with the storage key Kstm to memory card 40.

**[0081]** At step S22, memory card 40 decrypts the contents key CK with the storage key Kstm (namely, IDES (Kstm, DES (Kstm, CK))). At step S23, memory card 40 encrypts the decrypted contents key with the session key SeK and sends DES (SeK, CK) to recorder/player 1.

**[0082]** At step S24, recorder/player 1 decrypts the contents key with the session key SeK. At step S25, recorder/player 1 creates a block key BK with the decrypted contents key CK, a part key PK, and a block seed BK\_SEED. At step S26, recorder/player 1 decrypts each encrypted part data area 102 with the block key BK block by block. The audio encoder/decoder7 decodes the decrypted audio data.

**[0083]** With reference to interface 11 shown in FIG. 2, FIG. 14 shows a timing chart of data being read from memory card 40. In other than state 0 (initial state), a clock signal used to synchronize data is sent through clock line SCK. When data is sent or received between recorder/player 1 and memory card 40, the signal level of status line SBS is low. An initial condition may be referred to as state 0 (initial state). At timing t31, recorder/player 1 causes the signal level of status line SBS to become high (state 1).

**[0084]** When the signal level of status line SBS becomes high, memory card 40 (S/P and P/S IF block 43) determines that state 0 has changed to state 1. In state 1, recorder/player 1 sends a read command to memory card 40 through data line DIO. Thus, memory card 40 receives the read command. The read command is a protocol command referred to as a Transfer Protocol Command ("TPC"). As will be described later, the protocol command designates the contents

of the communication and the length of data that follows.

**[0085]** At timing t32, after a command has been transmitted, the signal level of status line SBS changes from high to low. Thus, state 1 changes to state 2. In state 2, a process designated by a command received by memory card 40 is performed. In reality, data of an address designated by the read command is read from flash memory 42 to page buffer 45. While the process is being performed, a busy signal (high level) is sent to recorder/player 1 through data line DIO.

**[0086]** At timing t33, after data has been read from flash memory 42 to page buffer 45, the supplying of the busy signal is stopped. A ready signal (low level) that represents that memory card 40 is ready to send data in accordance with the read command is outputted to recorder/player 1.

**[0087]** When recorder/player 1 receives the ready signal from memory card 40, recorder/player 1 determines that memory card 40 is ready for processing the read command. At timing t34, recorder/player 1 causes the signal level of status line SBS to become high. In other words, state 2 changes to state 3.

**[0088]** In state 3, memory card 40 outputs data that has been read to page buffer 45 in state 2 to recorder/player 1 through data line DIO. At timing t35, after the read data has been sent, recorder/player 1 stops sending the clock signal through clock line SCK. In addition, recorder/player 1 causes the signal level of status line SBS to change from high to low. Thus, state 3 changes to the initial state (state 0).

**[0089]** When an interrupt process should be performed such as due to a state change in memory card 40 as at timing t36, memory card 40 sends an interrupt signal to recorder/player 1 through data line DIO. When recorder/player 1 receives the interrupt signal through data line DIO from memory card 40 in state 0, recorder/player 1 determines that the signal is an interrupt signal and performs a process corresponding to the interrupt signal.

**[0090]** FIG. 15 is a timing chart of an operation in which data is written to flash memory 42 of memory card 40. In the initial state (state 0), the clock signal is not sent through clock line SCK. At timing t41, recorder/player 1 causes the signal level of status line SBS to change from low to high. Thus, state 0 changes to state 1. In state 1, memory card 40 is ready to receive a command. At timing t41, a write command is sent to memory card 40 through data line DIO and memory card 40 receives the write command.

**[0091]** At timing t42, recorder/player 1 causes the signal level of status line SBS to change from high to low. Thus, state 1 changes to state 2. In state 2, recorder/player 1 sends write data to memory card 40 through data line DIO and memory card 40 stores the received write data to page buffer 45.

**[0092]** At timing t43, recorder/player 1 causes the signal level of status line SBS to change from low to high. Thus, state 2 changes to state 3. In state 3, memory card 40 writes the write data to flash memory 42, memory card 40 sends a busy signal (high level) to recorder/player 1 through data line DIO, and recorder/player 1 sends a write command to memory card 40. Since the current state is state 3, recorder/player 1 determines that the signal received from memory card 40 is a status signal.

**[0093]** At timing t44, memory card 40 stops outputting the busy signal and sends a ready signal (low level) to recorder/player 1. When recorder/player 1 receives the ready signal, recorder/player 1 determines that the writing process corresponding to the write command has been completed and stops sending the clock signal. Additionally at timing t45, recorder/player 1 causes the signal level of status line SBS to change from high to low. Thus, state 3 returns to state 0 (initial state).

**[0094]** When recorder/player 1 receives a high level signal from memory card 40 through data line DIO in state 0, recorder/player 1 determines that the received signal is an interrupt signal. Recorder/player 1 performs a process corresponding to the received interrupt signal. When memory card 40 is to be detached from recorder/player 1, memory card 40 generates the interrupt signal.

**[0095]** In other than the reading process and the writing process, in state 1, a command is sent. In state 2, data corresponding to the command is sent.

**[0096]** It is noted that the serial interface disposed between recorder/player 1 and memory card 40 is not limited to interface 11 as described above. In other words, various types of serial interfaces may be used.

**[0097]** FIG. 16 is a table depicting examples of protocol commands (TPC codes) sent through the data line DIO of the serial interface. The data length of each protocol command is one byte. In FIG. 16, each protocol command is represented in hexadecimal notation (with suffix h) and decimal notation (0 and 1). In addition, definitions of individual protocol commands are represented for both the non-security type memory card 40' (see FIG. 3) and the security type memory card 40 (see FIG. 2). In FIG. 16, R and W represent a read type protocol command and a write type protocol command, respectively. As described above, since a command is sent in state 1 and data is sent in state 2, the data length (in bytes) corresponding to each protocol command is shown.

**[0098]** At this point, each of the protocol commands TPC will be described.

**[0099]** TPC = 2Dh is an access command to a conventional flash memory (this command is simply referred to as memory control command). This command is a page data read command and is common to the memory cards 40 and 40'. The length of data preceded by the command is the data length for one page (512 bytes + 2 bytes (CRC)). The page data is read from the page buffer 45.

**[0100]** TPC = D2h is a memory control command. This command is a page data write command. The length of data

preceded by the command is the data for one page (512 bytes + 2 bytes (CRC)). The page data is written to the page buffer 45.

**[0101]** TPC = 4Bh is a memory control command. This command is a read command against the read register 48. The data length of data preceded by the command is (31 bytes + 2 bytes (CRC)).

**[0102]** TPC = B4h is a memory control command. This command is a write command against the write register 46. The data length of data preceded by the command is (31 bytes + 2 bytes (CRC)).

**[0103]** TPC = 78h is a memory control command. This command is a command for reading one byte from the read register 48. The data length of data preceded by the command is (1 byte + 2 bytes (CRC)).

**[0104]** TPC = 87h is a memory control command. This command is a command for varying the access range of the command register 44. The data length of data preceded by the command is (4 bytes + 2 bytes (CRC)).

**[0105]** TPC = 1Eh is a data read command for the status register of the security block 52 of the memory card 40. However, this command is not defined for the memory card 40'. The data length of data preceded by the command is (2 bytes + 2 bytes (CRC)). A command dedicated for the security block 52 is referred to as security command.

**[0106]** TPC = E1h is a memory control command. This command is a command set command against the command register 44. This command is followed by a command in a lower hierarchical level than TPC commands. Thus, the data length of this command is (1 byte + 2 bytes (CRC)).

**[0107]** TPC = 3Ch is a security data read command against the security block 52 of the memory card 40. However, this command is not defined for the memory card 40'. The data length of data preceded by the command is (24 bytes + 2 bytes (CRC)).

**[0108]** TPC = C3h is a security data write command against the security block 52 of the memory card 40. However, this command is not defined for the memory card 40'. The data length of data preceded by the command is (26 bytes + 2 bytes (CRC)).

**[0109]** With reference now to FIGS. 17 and 18, a command (1 byte) followed by the TPC = E1h command will be described. FIG. 17 shows commands for the non-security type memory card 40'. These are as follows:

E1h = AAh: block read command  
 E1h=55h: block write command  
 E1h=33h: block read/write cancel command  
 E1h = 99h: block erase command  
 E1h = CCh: memory operation stop command  
 E1h = 5Ah: power save mode command  
 E1h = C3h: page buffer clear command  
 E1h = 3Ch: memory controller reset command

**[0110]** FIG. 18 shows commands for the security type memory card 40. Since the definitions of the commands (AAh to 3Ch) shown in FIG. 18 are the same as those shown in FIG. 17, they are omitted. In other words, these commands are memory control commands defined in common with the memory cards 40 and 40'. In FIG. 18, commands (60h to 83h) are security commands for an encrypting process (including a decrypting process and an authenticating process) dedicated for the memory card 40.

**[0111]** As shown in FIGS. 17 and 18, the memory control commands TPC in common with the memory cards 40 and 40' and security commands TPC dedicated for the memory card 40 are defined. Likewise, this relation applies to commands in lower hierarchical levels. In other words, in the lower hierarchical levels, common memory control commands and security commands are defined. The security commands are not defined (not used) for the memory card 40'. According to the illustrative embodiment, when the S/P and P/S IF block 43 receives a command from the recorder 1 through the serial interface, the memory card 40 determines whether or not the received command TPC is a common memory control command or a security command. The memory card 40 sends subsequent data to an appropriate circuit corresponding to the determined result. When the received command is for example the TPC = E1h command of which a command is followed by another command, the memory card 40 sends the command to a proper circuit corresponding to the definitions for the commands shown in FIG. 18.

**[0112]** FIG. 19 depicts an arrangement for selecting a circuit to which data is intended for, in correspondence with a received command. The arrangement is embodied within interface circuit 43 of memory card 40. Data is sent from recorder 1 to memory card 40 through data line DIO. The received data is supplied to a terminal "a" of a switch circuit 152 through a delay circuit 150. In addition, the received data is supplied to an input terminal of a detecting circuit 151. Detecting circuit 151 determines whether or not a protocol command (TPC) received through the data line DIO is a memory control command or a security command, according to the code value of the protocol command. Switch circuit 152 is controlled in accordance with the determined result. Delay circuit 150 compensates the detecting time of detecting circuit 151. These structural elements are accomplished by hardware and/or software in the S/P and P/S IF block 43. According to the embodiment, since codes that are not used for memory control commands are assigned to security

commands, detecting circuit 151 can easily determine these two types of commands.

[0113] When the detecting circuit 151 has determined that the received protocol command is a memory control command, the terminal "a" of the switch circuit 151 is connected to a terminal "b". Thus, the memory control command is supplied to a page buffer (e.g., page buffer 45 shown in FIG. 2, but omitted in FIG. 19 for clarity), a register (e.g., register 46 or 48 shown in FIG. 2), and so forth through the terminals "a" and "b" of the switch circuit 151 so as to control the flash memory 42. Data following the memory control command is supplied to the page buffer, the register, and so forth. Alternatively, data is sent from the page buffer, the register, and so forth to the recorder 1 through the terminals "b" and "a" of the switch circuit 151.

[0114] When the detecting circuit 151 has determined that the received protocol command is a security command, the terminal "a" of the switch circuit 151 is connected to a terminal "c" thereof. The security command is supplied to the security block 52 through the terminals "a" and "c" of the switch circuit 151. Data following the security command is supplied to the security block 52. The data is sent from security block 52 to recorder 1 through the terminals "a" and "c" of switch circuit 151.

[0115] When the received command is the protocol command (TPC = E1h), it is followed by a normal memory control command or a security command. When the detecting circuit 151 receives the TPC = E1h protocol command, the detecting circuit 151 determines whether the command is followed by a control command or a security command. Memory card 40 then controls the switch circuit 151 according to the determined result. When the received command is other than the command TPC = E1h and it is followed by a memory control command or a security command, the memory card 40 can send data to a proper circuit corresponding to the code value of the command.

[0116] Since memory card 40 has a function for determining whether the received command is a memory control command or a security command, memory card 40 can be used for a non-security type recorder. In other words, a non-security type recorder does not exchange security information with memory card 40. The non-security type recorder sends only write/read memory control commands and data corresponding thereto to memory card 40. As described above, memory card 40 determines whether or not a command received from a recorder is a memory control command and writes or reads data corresponding thereto to/from the flash memory 42. Thus, data can be written or read to/from the memory card 40.

[0117] According to the above-described embodiment, DES was described as a preferred encrypting method. It is understood, however, that various other encrypting technologies can be used in the alternative.

[0118] According to the present invention, a memory card having a non-volatile memory and a security block can be used with both security type and non-security type data processing units (electronic units) such as audio and/or video recorders. Thus, the compatibility of a security type memory card is improved.

[0119] In addition, according to the present invention, since codes that are not used in the communication between a data processing unit and a memory card are assigned for control data for a security operation, the above-noted compatibility is obtained against a conventional non-security type memory card without any disadvantage. In other words, when a non-security type electronic unit is available, a security type memory card according to the present invention can be used with the electronic unit. In a method for adding a new identifier to data exchanged between an electronic unit and a memory card and identifying the data type, in addition to the requirement of the new identifier, a conventional electronic unit cannot be used. However, with the present invention, which does not exhibit this problem, compatibility with a conventional electronic unit and a conventional memory card can be achieved.

[0120] It is also to be understood that the following claims are intended to cover all of the generic and specific features of the invention herein described and all statements of the scope of the invention which, as a matter of language, might be said to fall therebetween.

## Claims

1. A memory unit removably attachable to and operational with a data processing unit (1), said memory unit (40) comprising a non-volatile memory (42) and being **characterised by**:

security means (52) for protecting the security of data stored in said non-volatile memory (42); and an interface (43) for receiving, from the data processing unit (1), control data, **characterized in that** the interface receives first control data and second control data different from said first control data, said interface supplying received first control data for a read or write operation with respect to said non-volatile memory (42), and supplying received second control data for a security operation of said security means (52);

the memory unit (40) being removably attachable to and operational with a non-security type data processing unit that transmits said first control data and does not transmit said second control data, and also being removably attachable to and operational with a security type data processing unit the transmits both said first and second control data.

2. A memory unit as set forth in claim 1, wherein said interface (43) comprises a detection means (151) for detecting whether incoming control data from said data processing unit (1) is said first control data or said second control data, and a switching means (152) for switching said control data, in accordance with the detection of said detection means (151), in such a manner that said first control data is supplied to said non-volatile memory (142) and said second control data is supplied to said security means (152).
3. A memory unit as set forth in claim 1 or claim 2, wherein following the reception by said interface of said first or second control data, said interface (43) is operable to receive data defined by the respective first or second control data.
4. A memory unit as set forth in claim 3, wherein said data that said interface (43) is operable to receive after said first control data or said second control data includes a first command for a reading or writing operation for said non-volatile memory (42) and a second command, different from said first command, for a security operation of said security means (52).
5. A memory unit as set forth in claim 4, wherein said interface (43) is operable to supply said first command for reading or writing operation for said non-volatile memory (42) and supply said second command for a security operation of said security means (52).
6. A memory unit as set forth in any one of the preceding claims, wherein said interface (43) is operable to output said first control data to at least one of a page buffer (45), a write register (46) and a read register (48) operatively coupled between said interface (43) and said non-volatile memory (42).
7. A memory unit as set forth in any one of the preceding claims, which is removably attachable to and operational with non-security and security type data processing units that are audio recorders/players or image recording/reproducing devices.
8. A memory unit as set forth in any one of the preceding claims, wherein said security means (52) is configured to protect security of data stored in the non-volatile memory (42) in association with security means (3) of said data processing unit (1) by sharing a session key.
9. A data processing system comprising a data processing unit (1) and a memory unit (40), as set forth in any one of the preceding claims, removably attached to and operational with said data processing unit.
10. A data processing method for use in a data processing system having a data processing unit (1) and a memory unit (40) removably attachable to and operational with said data processing unit, said memory unit (40) comprising a non-volatile memory (42), the method being **characterised by:**

attaching the memory unit (40) to a security type data processing unit that can transmit both first control data for a reading or writing operation with respect to the non-volatile memory (42), and second control data, which is different from said first control data for a security operation of a security means (52) provided in the memory unit (40) for protecting the security data stored in said non-volatile memory (42), and attaching the memory unit (40) to a non-security type data processing unit that can transmit said first control data but not said second control data;

transmitting, from the data processing unit (1) to the memory unit (40), said first and/or second control data; and receiving said transmitted control data at an interface (43) of said memory unit (40), determining whether it is first or second control data, and supplying received first control data to said non-volatile memory (42) and received second control data to said security means (52).

## Patentansprüche

1. Speichereinheit, die lösbar mit einer Datenverarbeitungseinheit (1) verbindbar und mit dieser betreibbar ist, wobei die Speichereinheit (40) einen nicht-flüchtigen Speicher (42) umfasst und **gekennzeichnet ist durch:**

eine Sicherheitseinrichtung (52) zum Schützen der Sicherheit von in dem nicht-flüchtigen Speicher (42) gespeicherten Daten; und

eine Schnittstelle (43) zum Empfangen von Steuerdaten aus der Datenverarbeitungseinheit (1), **dadurch ge-**

**kennzeichnet, dass** die Schnittstelle erste Steuerdaten und zweite Steuerdaten, die von den ersten Steuerdaten verschieden sind, empfängt, wobei die Schnittstelle die empfangenen ersten Steuerdaten für eine Lese- oder Schreiboperation mit Bezug auf den nicht-flüchtigen Speicher (42) bereitstellt und die empfangenen zweiten Steuerdaten für eine Sicherheitsoperation der Sicherheitseinrichtung (52) bereitstellt;

wobei die Speichereinheit (40) mit einer nicht sicheren Datenverarbeitungseinheit lösbar verbindbar und betreibbar ist, die erste Steuerdaten und nicht die zweiten Steuerdaten überträgt, und auch mit einer sicheren Datenverarbeitungseinheit lösbar verbindbar und betreibbar ist, die sowohl die ersten als auch die zweiten Steuerdaten überträgt.

2. Speichereinheit nach Anspruch 1, wobei die Schnittstelle (43) eine Detektionseinrichtung (151) umfasst, um zu detektieren, ob von der Datenverarbeitungseinheit (1) eintreffende Steuerdaten ersten Steuerdaten oder zweiten Steuerdaten entsprechen, und eine Umschalteneinrichtung (152) zum Schalten der Steuerdaten gemäß dem Detektionsergebnis der Detektionseinrichtung (151) umfasst, so dass die ersten Steuerdaten dem nicht-flüchtigen Speicher (142) zugeführt werden, und so dass die zweiten Steuerdaten der Sicherheitseinrichtung (152) zugeführt werden.
3. Speichereinheit nach Anspruch 1 oder 2, wobei die Schnittstelle (43) ausgebildet ist, um nach dem Empfang der ersten oder zweiten Steuerdaten durch die Schnittstelle die Daten, die jeweils durch die ersten und die zweiten Steuerdaten definiert sind, zu empfangen.
4. Speichereinheit nach Anspruch 3, wobei die Schnittstelle (43) ausgebildet ist, um nach den ersten Steuerdaten oder den zweiten Steuerdaten einen ersten Befehl für eine Lese- oder Schreiboperation für den nicht-flüchtigen Speicher (42) und einen zweiten Befehl, der von dem ersten Befehl verschieden ist, für eine Sicherheitsoperation der Sicherheitseinrichtung (52) zu empfangen.
5. Speichereinheit nach Anspruch 4, wobei die Schnittstelle (43) ausgebildet ist, um einen ersten Befehl für die Lese- oder Schreiboperation für den nicht-flüchtigen Speicher (42) und den zweiten Befehl für eine Sicherheitsoperation der Sicherheitseinrichtung (52) bereitzustellen.
6. Speichereinheit nach einem der vorangehenden Ansprüche, wobei die Schnittstelle (43) ausgebildet ist, um die ersten Steuerdaten an einen Seitenpuffer (45) und/oder ein Schreibregister (46) und/oder ein Leseregister (48) auszugeben, die zwischen der Schnittstelle (43) und dem nicht-flüchtigen Speicher (42) funktional miteinander verbunden sind.
7. Speichereinheit nach einem der vorangehenden Ansprüche, die mit der nicht sicheren Datenverarbeitungseinheit und mit der sicheren Datenverarbeitungseinheit lösbar verbindbar und betreibbar sind, die Audio-Aufzeichnungsgeräten/-Wiedergabegeräten oder Bildaufzeichnungsgeräten/-Wiedergabegeräten entsprechen.
8. Speichereinheit nach einem der vorangehenden Ansprüche, wobei die Sicherheitseinrichtung (52) ausgebildet ist, um die Sicherheit von in dem nicht-flüchtigen Speicher (42) gespeicherten Daten gemäß der Sicherheitseinrichtung (3) der Datenverarbeitungseinheit (1) durch Austauschen eines Sitzungsschlüssels zu schützen.
9. Datenverarbeitungssystem, das eine Datenverarbeitungseinheit (1) und eine Speichereinheit (40) nach einem der vorangehenden Ansprüche umfasst, die mit der Datenverarbeitungseinheit lösbar verbindbar und betreibbar ist.
10. Datenverarbeitungsverfahren zur Verwendung in einem Datenverarbeitungssystem mit einer Datenverarbeitungseinheit (1) und einer Speichereinheit (40), die mit der Datenverarbeitungseinheit lösbar verbindbar und betreibbar ist, wobei die Speichereinheit (40) einen nicht-flüchtigen Speicher (42) umfasst, mit folgenden Schritten:

Verbinden der Speichereinheit (40) mit einer sicheren Datenverarbeitungseinheit, die sowohl erste Steuerdaten für eine Lese- oder Schreiboperation mit Bezug auf den nicht-flüchtigen Speicher (42) übertragen kann als auch zweite von den ersten Steuerdaten verschiedene Steuerdaten für eine Sicherheitsoperation einer Sicherheitseinrichtung (52), die in der Speichereinheit (40) vorgesehen ist, zum Schützen der Sicherheitsdaten, die in dem nicht-flüchtigen Speicher (42) gespeichert sind, übertragen kann, und Verbinden der Speichereinheit (40) mit einer nicht sicheren Datenverarbeitungseinheit, die die ersten Steuerdaten übertragen kann, jedoch nicht die zweiten Steuerdaten;

Übertragen der ersten und zweiten Steuerdaten von der Datenverarbeitungseinheit (1) an die Speichereinheit (40); und

Empfangen der übertragenen Steuerdaten an eine Schnittstelle (43) der Speichereinheit (40); Bestimmen, ob

diese ersten oder zweiten Steuerdaten entsprechen, und Zuführen der empfangenen ersten Steuerdaten an den nicht-flüchtigen Speicher (42) und der empfangenen zweiten Steuerdaten an die Sicherheitseinrichtung (52).

## 5 Revendications

1. Unité de mémoire pouvant être fixée de façon amovible à et opérationnelle avec une unité de traitement de données (1), ladite unité de mémoire (40) comportant une mémoire non volatile (42) et étant **caractérisée par** :

un moyen sécurisé (52) pour protéger la sécurité des données mémorisées dans ladite mémoire non volatile (42); et

une interface (43) pour recevoir, depuis l'unité de traitement de données (1), des données de contrôle, **caractérisée en ce que** l'interface reçoit des premières données de contrôle et des secondes données de contrôle différentes desdites premières données de contrôle, ladite interface délivrant les premières données de contrôle reçues pour une opération de lecture ou d'écriture par rapport à ladite mémoire non volatile (42), et délivrant les secondes données de contrôle reçues pour une opération de sécurité dudit moyen sécurisé (52) ;

l'unité de mémoire (40) pouvant être fixée de façon amovible à et opérationnelle avec une unité de traitement de données de type non sécurisées qui transmet lesdites premières données de contrôle et ne transmet pas lesdites secondes données de contrôle, et pouvant être également fixée de façon amovible à et opérationnelle avec une unité de traitement de données de type sécurisées qui transmet à la fois lesdites premières et les secondes données de contrôle.

2. Unité de mémoire selon la revendication 1, dans laquelle ladite interface (43) comporte des moyens de détection (151) pour détecter si les données de contrôle provenant de ladite unité de traitement de données (1) sont lesdites premières données de contrôle ou lesdites secondes données de contrôle, et des moyens de commutation (152) pour commuter lesdites données de contrôle, conformément à la détection desdits moyens de détection (151), de telle manière que lesdites premières données de contrôle sont délivrées à ladite mémoire non volatile (42) et lesdites secondes données de contrôle sont délivrées audit moyen sécurisé (52).

3. Unité de mémoire selon la revendication 1 ou 2, dans laquelle, à la suite de la réception par ladite interface desdites premières ou secondes données de contrôle, ladite interface (43) est capable de recevoir les données définies par lesdites premières ou secondes données de contrôle.

4. Unité de mémoire selon la revendication 3, dans laquelle lesdites données que ladite interface (43) est capable de recevoir après que lesdites premières données de contrôle ou lesdites secondes données de contrôle comprennent une première commande pour une opération de lecture ou d'écriture pour ladite mémoire non volatile (42) et une seconde commande, différente de ladite première commande, pour une opération de sécurité dudit moyen sécurisé (52).

5. Unité de mémoire selon la revendication 4, dans laquelle ladite interface (43) est capable de délivrer ladite première commande pour une opération de lecture ou d'écriture pour ladite mémoire non volatile (42) et délivrer ladite seconde commande pour une opération de sécurité dudit moyen sécurisé (52).

6. Unité de mémoire selon l'une quelconque des revendications précédentes, dans laquelle ladite interface (43) peut sortir lesdites premières données de contrôle à au moins l'un d'une mémoire tampon de page (45), d'un registre d'écriture (46) et d'un registre de lecture (48) couplé fonctionnellement entre ladite interface (43) et ladite mémoire non volatile (42).

7. Unité de mémoire selon l'une quelconque des revendications précédentes, qui est fixée de façon amovible à et opérationnelle avec des unités de traitement de données de type non sécurisées et sécurisées qui sont des enregistreurs/lecteurs audio ou des dispositifs d'enregistrement/reproduction d'images.

8. Unité de mémoire selon l'une quelconque des revendications précédentes, dans laquelle ledit moyen sécurisé (52) est configuré pour protéger la sécurité des données mémorisées dans la mémoire non volatile (42) en association avec un moyen sécurisé (3) de ladite unité de traitement de données (1) en partageant une clé de session.

9. Système de traitement de données comportant une unité de traitement de données (1) et une unité de mémoire (40), selon l'une quelconque des revendications précédentes, fixée de façon amovible à et opérationnelle avec



ladite unité de traitement de données.

- 5 10. Procédé de traitement de données utilisé dans un système de traitement de données possédant une unité de traitement de données (1) et une unité de mémoire (40) pouvant être fixée de façon amovible à et opérationnelle avec ladite unité de traitement de données, ladite unité de mémoire (40) comportant une mémoire non volatile (42), procédé **caractérisé par** :

10 la fixation de l'unité de mémoire (40) à une unité de traitement de données de type sécurisées qui peut transmettre à la fois les premières données de contrôle pour une opération de lecture ou d'écriture par rapport à la mémoire non volatile (42), et les secondes données de contrôle qui sont différentes desdites premières données de contrôle pour une opération de sécurité du moyen sécurisé (52) prévu dans l'unité de mémoire (40) pour protéger les données sécurisées mémorisées dans ladite mémoire non volatile (42), et la fixation de l'unité de mémoire (40) à une unité de traitement de données de type non sécurisées qui peut transmettre lesdites premières données de contrôle mais pas lesdites secondes données de contrôle ;

15 la transmission, depuis l'unité de traitement de données (1) à l'unité de mémoire (40), desdites premières et/ou secondes données de contrôle ; et

20 la réception desdites données de contrôle transmises à une interface (43) de ladite unité de mémoire (40), déterminant si ce sont les premières ou secondes données de contrôle, et délivrant les premières données de contrôle reçues à ladite mémoire non volatile (42) et les secondes données de contrôle reçues audit moyen sécurisé (52).

25

30

35

40

45

50

55

Fig. 1

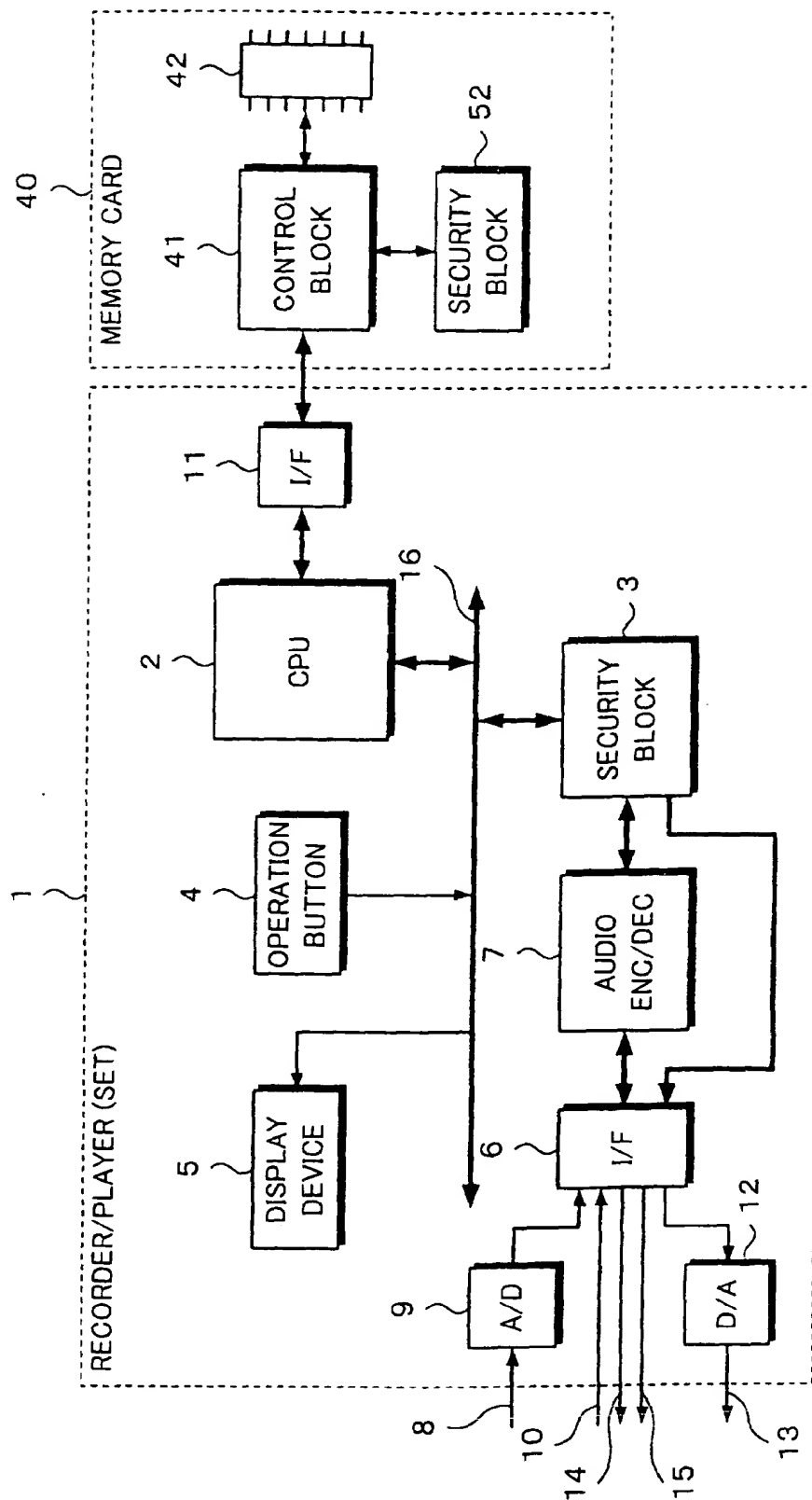


Fig. 2

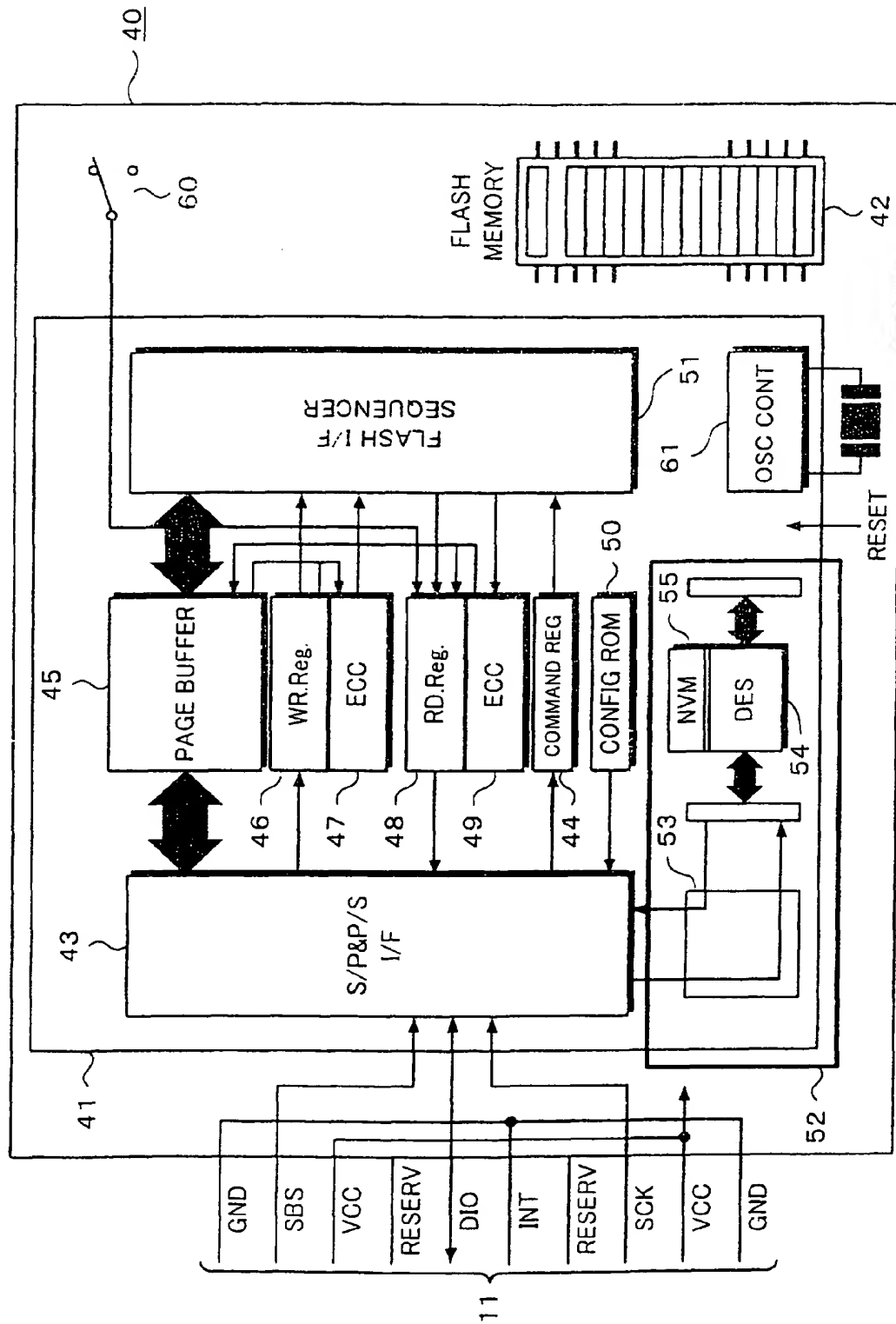
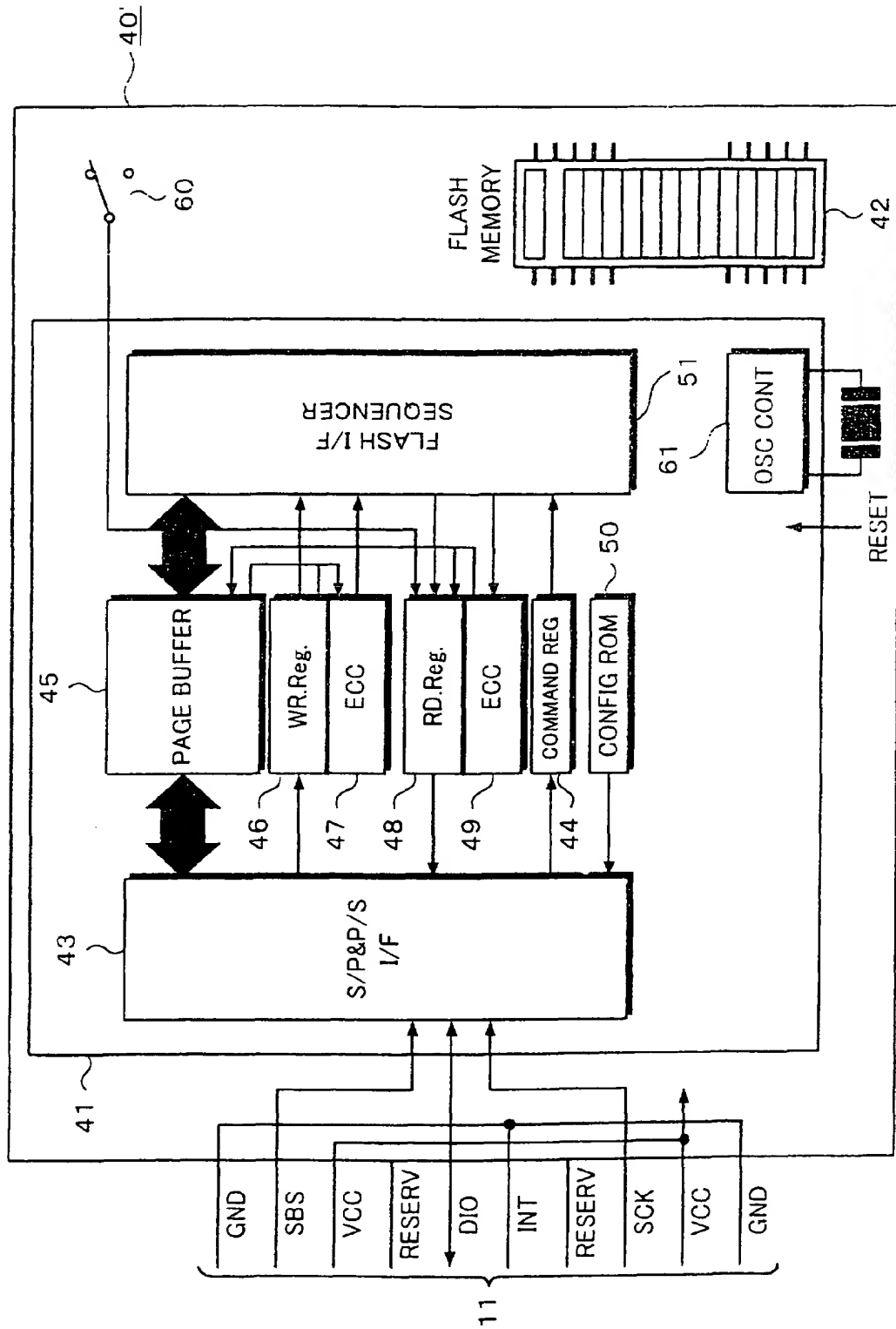
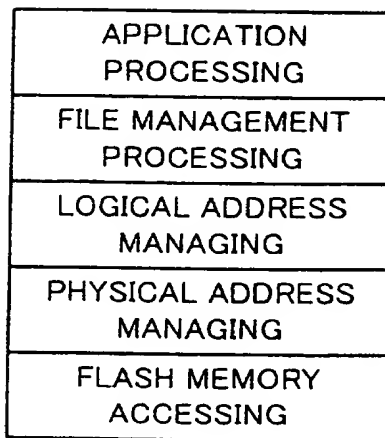


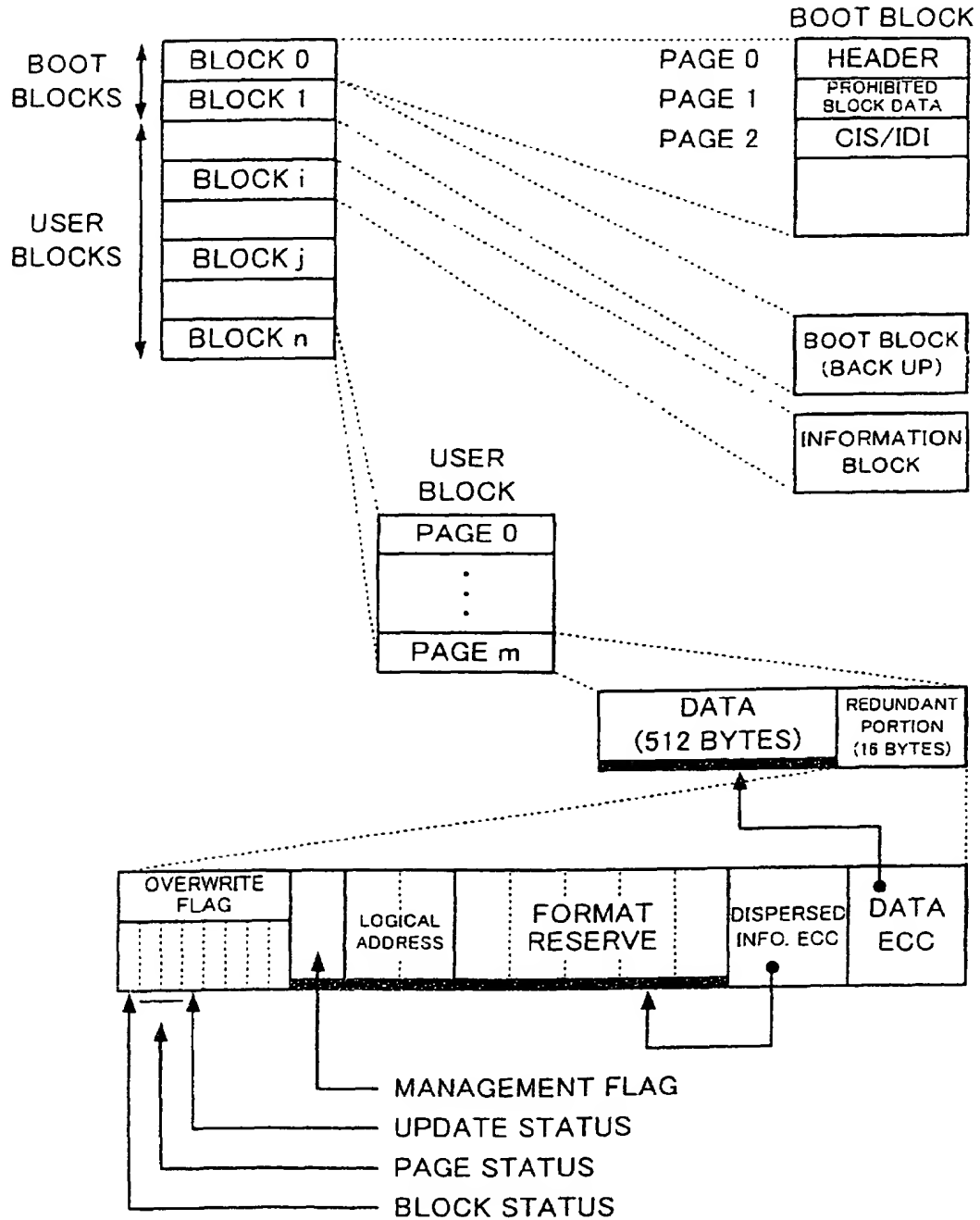
Fig. 3

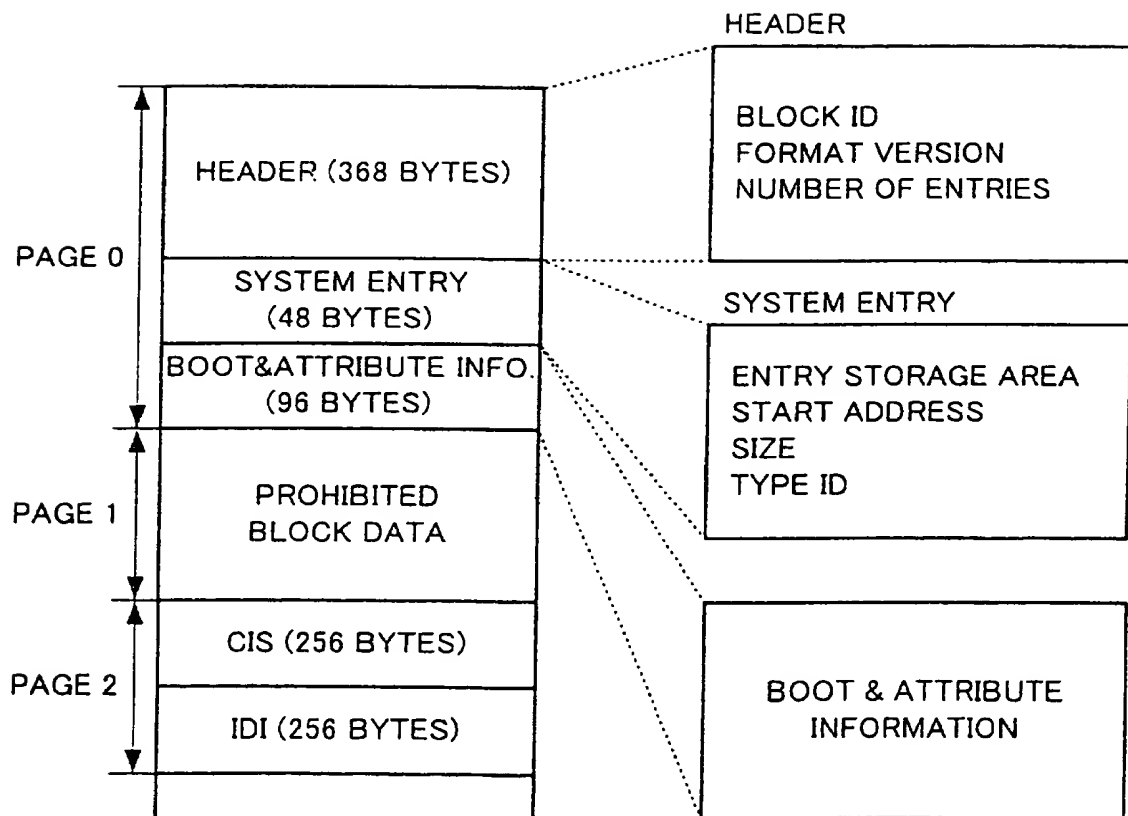


*Fig. 4*



FILE SYSTEM PROCESSING  
HIERARCHY

*Fig. 5*

*Fig. 6*

*Fig. 7*

		NUMBER OF BYTES	
MS CLASS (*)	1	1	1. TYPE-1 OTHER RESERVED
CARD TYPE (*)	1	1	1. READ ONLY 2. READ WRITE 3. HYBRID OTHER RESERVED
BLOCK SIZE (*)	2	2	BLOCK SIZE IN KB 16KB 0x0010 8KB 0x0008
NUMBER OF BLOCKS (*)	2	2	NUMBER OF BLOCKS
TOTAL NUMBER OF BLOCKS (*)	2	2	TOTAL NUMBER OF BLOCKS
PAGE SIZE	2	2	PAGE SIZE. 512 FIXED. 0x0200
SIZE OF REDUNDANT PORTION	1	1	SIZE OF REDUNDANT PORTION =16 BYTES. 0x10
SECURITY TYPE (*)	1	1	
DATE AND TIME OF ASSEMBLY (*)	8	8	DATE OF PRODUCTION OF CARD (HARD) (SEE DATE AND TIME DESIGNATION FORMAT ON NEXT PAGE)
MAKER AREA (*)	4	4	USED FOR MANAGEMENT IN MAKER SUCH AS SERIAL NUMBER
MS ASSEMBLY MAKER CODE (*)	1	1	REGISTERED ASSEMBLY MAKER CODE
MS ASSEMBLY TYPE CODE (*)	3	3	REGISTERED ASSEMBLY TYPE CODE
MEMORY MAKER CODE	2	2	CHIP MAKER CODE 0 : UNKNOWN
MEMORY DEVICE CODE	2	2	DEVICE CODE 0 : UNKNOWN
MEMORY SIZE	2	2	MB ex : 32 MBITS FLASH 0x0004
FORMAT RESERVE	1	1	1 : OTHER RESERVED
FORMAT RESERVE	1	1	1 : OTHER RESERVED
VCC	1	1	VCC UNIT : 0.1 V ex ) 3.3 V 0x21
VPP	1	1	VPP UNIT : 0.1 V ex ) 3.3 V 0x21
CONTROLLER NUMBER	2	2	CONTROLLER CHIP NUMBER
RESERVE	14	14	
FORMAT TYPE (*)	1	1	1 : FAT OTHER RESERVED
APPLICATION	1	1	0 : GENERAL PURPOSE OTHER RESERVED
ZERO RESET RESERVE	5	5	
RESERVE	35	35	



Fig. 8A

CONTENTS IN MEMORY CARD

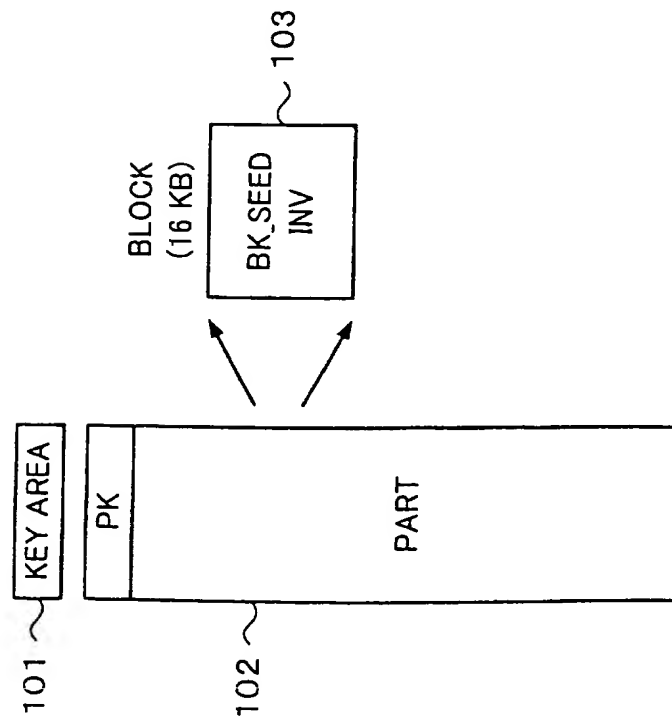


Fig. 8B

CONTENTS IN RECORDER

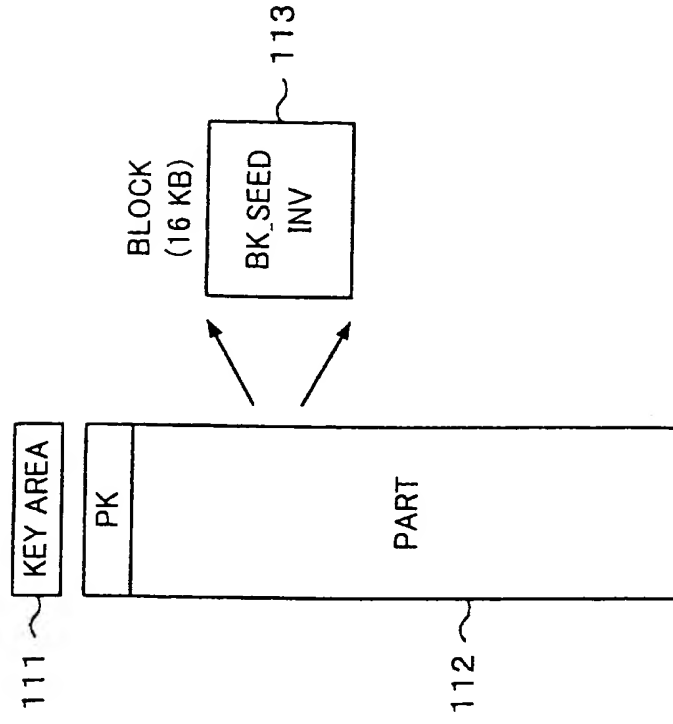
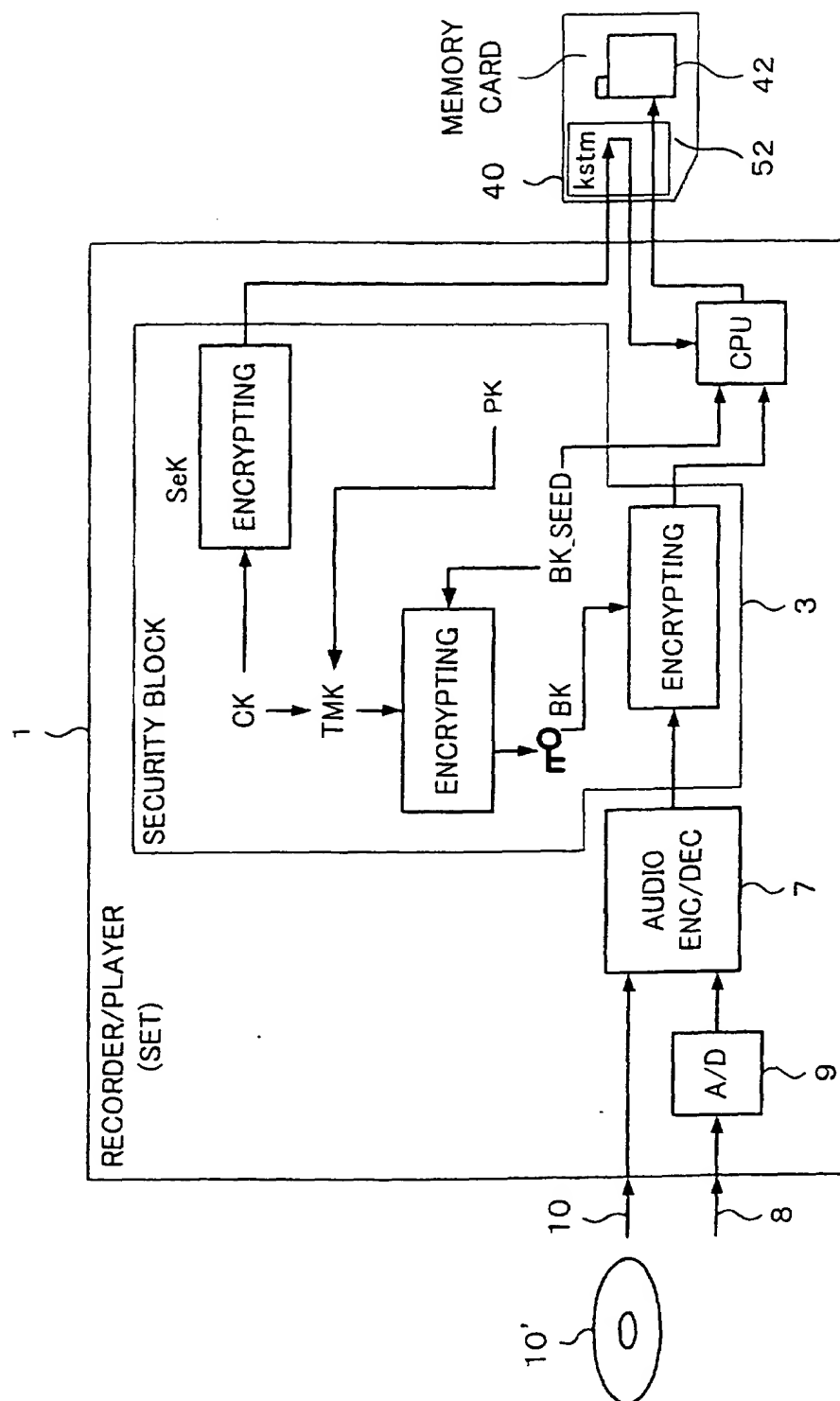
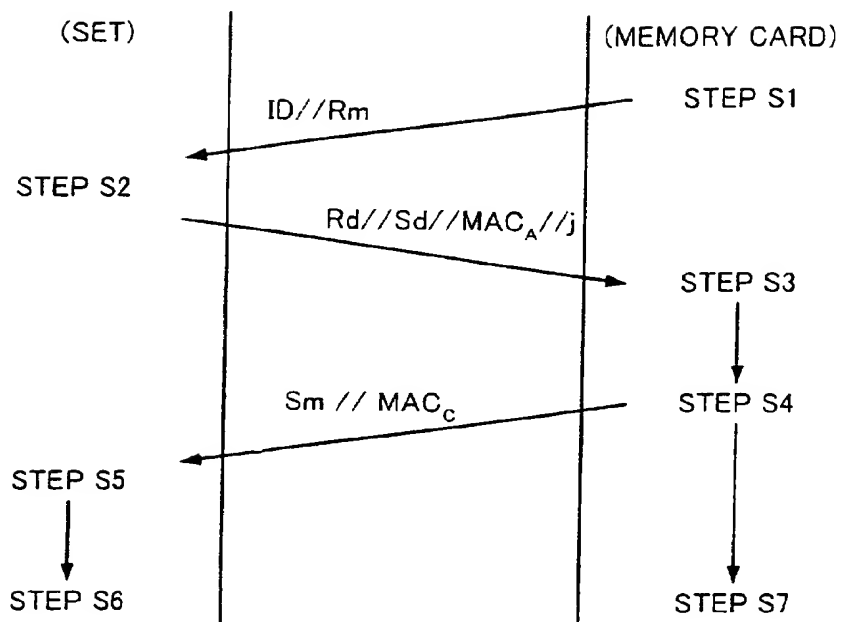


Fig. 9



**Fig. 10**



**Fig. 11**

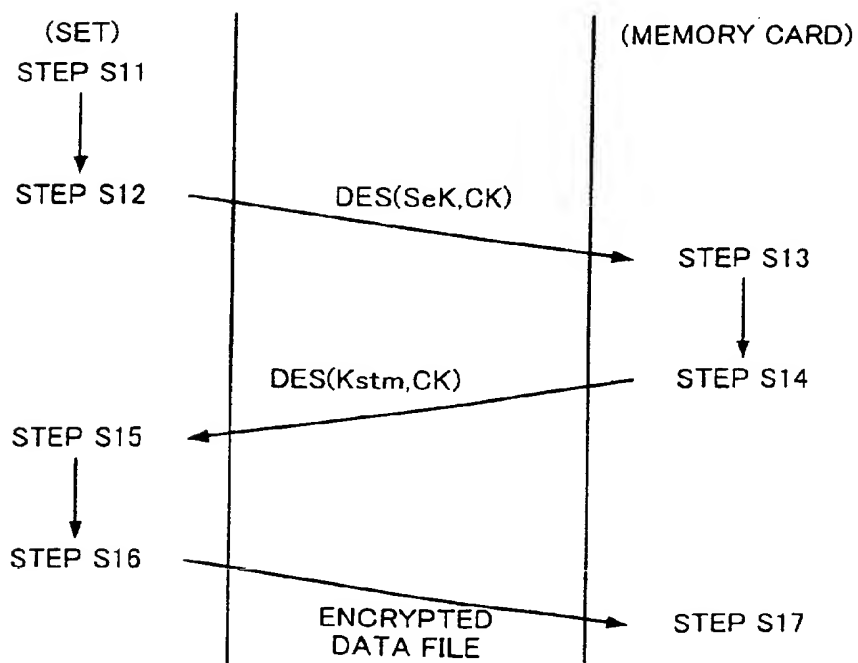
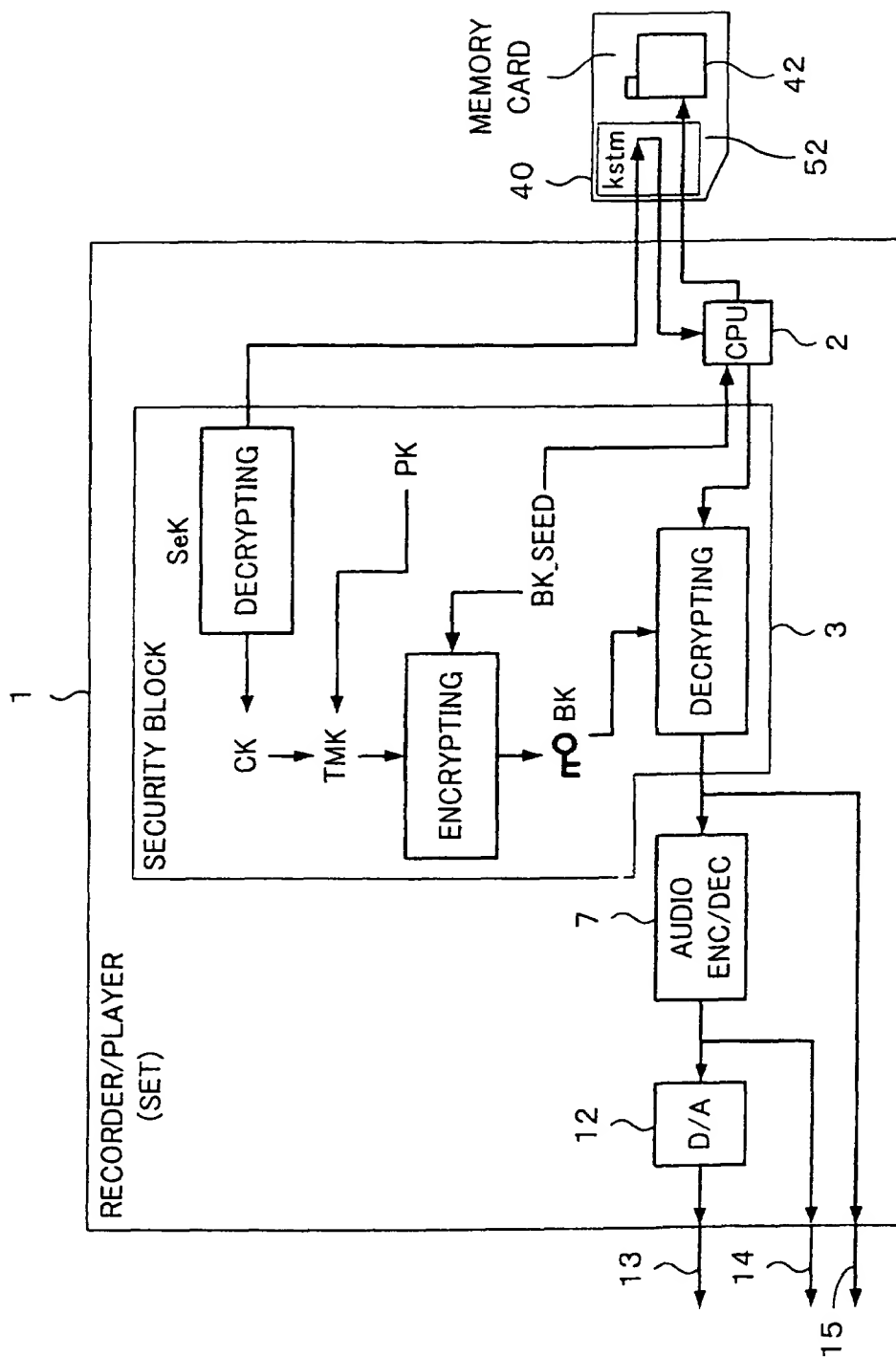
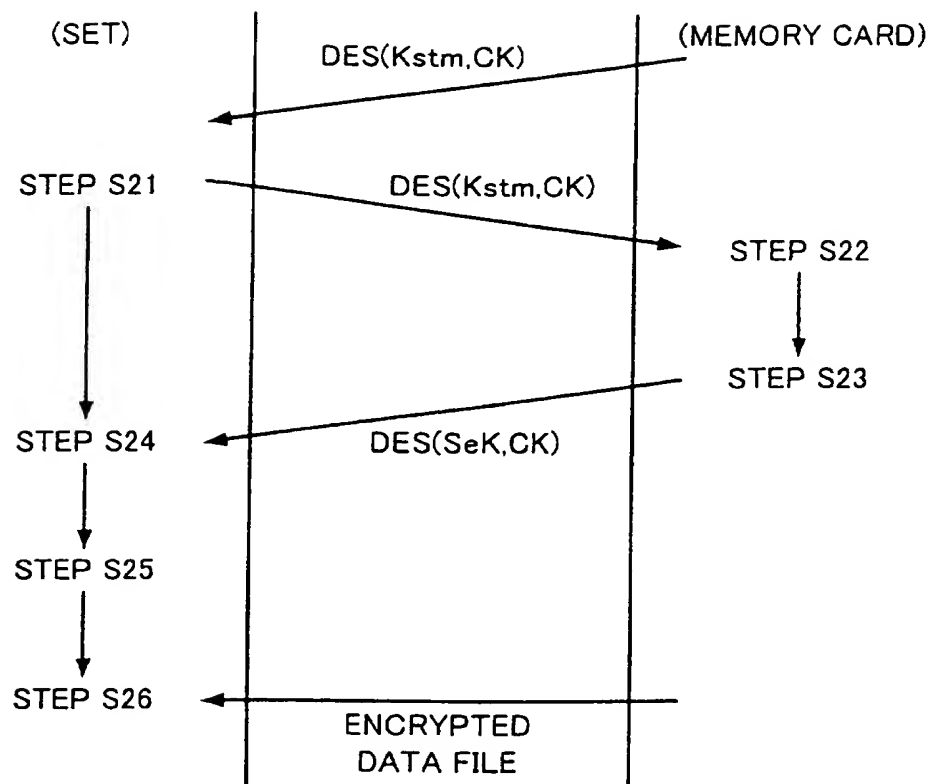
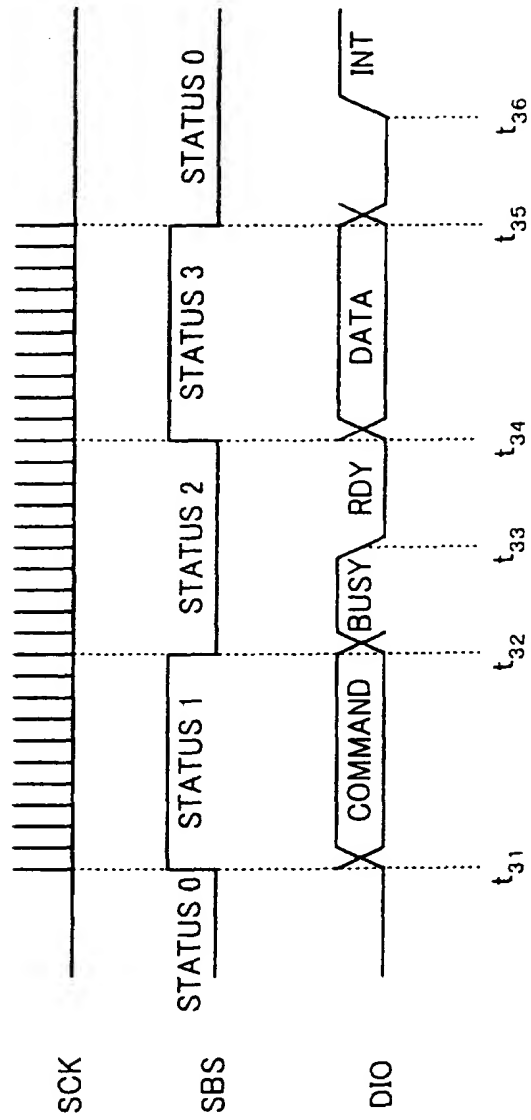


Fig. 12



**Fig. 13**



**Fig. 14**

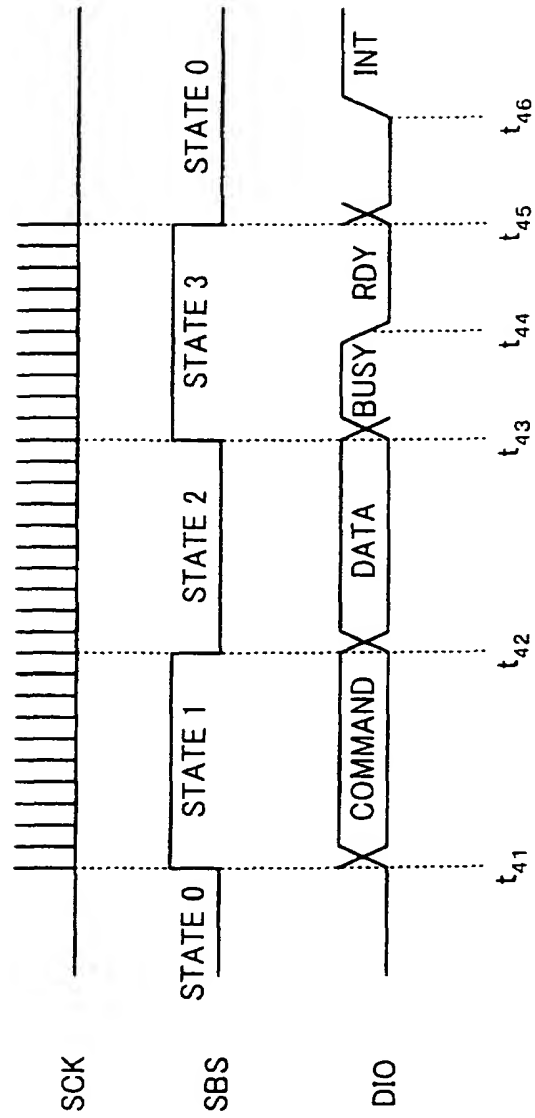
**Fig. 15**

Fig. 16

TPC		DEFINITION	DEFINITION	R/W	DATA LENGTH
2Dh	0010 1101	READ_PAGE_DATA	→	R	512B+2B(CRC)
D2h	1101 0010	WRITE_PAGE_DATA	→	W	512B+2B(CRC)
4Bh	0100 1011	READ_REG	→	R	31B+2B(CRC)
B4h	1011 0100	WRITE_REG	→	W	15B+2B(CRC)
78h	0111 1000	GET_INT	→	R	1B+2B(CRC)
87h	1000 0111	SET_REG_ADRS	→	W	4B+2B(CRC)
1Eh	0001 1110	Reserv	READ_STTS_REG	R	2B+2B(CRC)
E1h	1110 0001	SET-CMD	→	W	1B+2B(CRC)
3Ch	0011 1100	Reserv	READ_SSM_DATA	R	24B+2B(CRC)
C3h	1100 0011	Reserv	WRITE_SSM_DATA	W	26B+2B(CRC)



*Fig. 17*

TPC	CODE	DEFINITION
E1h	AAh	BLOCK_READ
	55h	BLOCK_WRITE
	33h	BLOCK_END
	99h	BLOCK_ERASE
	CCh	STOP
	5Ah	SLEEP
	C3h	CLEAR_BUF
	3Ch	RESET

*Fig. 18*

TPC	CODE	DEFINITION	CODE	DEFINITION
E1h	AAh	BLOCK_READ		
	55h	BLOCK_WRITE		
	33h	BLOCK_END		
	99h	BLOCK_ERASE		
	CCh	STOP		
	5Ah	SLEEP		
	C3h	CLEAR_BUF		
	3Ch	RESET		
	60h	LOAD_ID_CMD	72h	SET_KREC_CMD
	61h	SET_Rm_CMD	73h	MK_KREC_CMD
	62h	MK_Rm_CMD	74h	LOAD_KREC_CMD
	63h	LOAD_Rm_CMD	75h	SET_KPB_CMD
	64h	LOAD_MAC1D_CMD	76h	MK_KPB_CMD
	65h	SET_MAC1M_CMD	77h	LOAD_KPB_CMD
	66h	MK_MAC1M_CMD	78h	CLR_DEC_CMD
	67h	LOAD_MAC1M_CMD	79h	SET_ICV_CMD
	68h	CMP_CMD	7Ah	MK_ICV_CMD
	69h	MK_MAC2M_CMD	7Bh	LOAD_ICV_CMD1
	6Ah	LOAD_MAC2M_CMD	7Ch	LOAD_ICV_CMD2
	6Bh	SET_Sm_CMD	7Dh	LOAD_ICV_CMD3
	6Ch	MK_Sm_CMD	7Eh	LOAD_ICV_CMD4
	6Dh	LOAD_Sm_CMD	7Fh	CMP_ICV_CMD
	6Eh	SET_SeK_CMD	80h	LOAD_NVM_CMD
	6Fh	MK_SeK_CMD	81h	ALLEW_NVM_CMD
	70h	LOAD_SeK_CMD	82h	WR_NVM_CMD
	71h	CLR_IK_CMD	83h	RD_NVM_CMD

*Fig. 19*

